Collapsing Adverse Event Records

Haiqiang Luo, PPD Inc., Beijing

ABSTRACT

In clinical trials, the way of collecting adverse events (AEs) is diverse and collapsing AEs sometimes becomes necessary. Categorizing AEs into collapsible and non-collapsible, and collapsing the collapsible AEs are challenging tasks for SAS programmers in the clinical trials. This paper will introduce how to categorize AEs into collapsible and non-collapsible. Three types of collapsible AEs will be discussed. 1) Multiple AEs with same onset date; 2) Multiple AEs with a time contiguous sequence; 3) Overlapping AEs. This paper will introduce the implementation of collapsing and sample SAS codes for each type of collapsible AE records.

INTRODUCTION

An AE is defined as any new untoward medical occurrence or worsening of a pre-existing medical condition in a patient or clinical investigation subject administered an investigational (medicinal) product and that does not necessarily have a causal relationship with this treatment. An AE can therefore be any unfavorable and unintended sign (including an abnormal laboratory finding, for example), symptom, or disease temporally associated with the use of investigational product, whether or not considered related to the investigational product.

In clinical trial studies, the investigators are responsible for recording the patient’s AE at each clinical office visit or assessment. These AE records are stored in a clinical trial study database. The AE records are identified as either collapsible or non-collapsible.

- The first type is a set of multiple records with the same AE onset date. If the AE record has a missing end date, this indicates the event is continuing.
- The second type shows the records with a time contiguous sequence. It means the onset date of an event record is either the same day or 1 day later than the resolution date of a preceding event record. A typical instruction in a CRF for collecting this type of AE records are as follows:

  *If, at any time, the intensity increases or the Relationship to Study Drug changes, a Resolution Date for the ongoing event must be provided and the event re-recorded with the new information including the Onset Date/Time.*

- The third type shows the records with period overlapping. It means the onset date of an event record is after the onset date and prior to the resolution date of a preceding event record. This paper provides SAS codes for AE records collapsing, for each type of collapsible AE records.

  The codes use SQL procedure to separate AE to multiple AE records and non-multiple records, then separate multiple AE records to collapsible and non-collapsible AE, and collapse the collapsible records. At the last step of collapsing program, the collapsed records and non-collapsible data are set together.

In some cases, the following variables; AESER(Event Type, SAE/AE), AESEV(Severity/Intensity), AEREL(Relationship to Drug), AEACN(Action Taken), AEOUT(Outcome), AECONTRT(Concomitant or Add. Treatment Given) are not consistent for collapsing records. These variables are worst case related variables. The codes implement worst case scenario and select the worst values for the worst case related variables.
Figure 1 shows the flow chart of collapsing algorithm.

Source AE

No

Multiple AE

Yes

Same Onset Date

Yes

Collapse Records

No

Overlap AE

Yes

Collapse Records

No

Multiple AE

Yes

Contiguous AE

Yes

Collapse Records

No

Final AE
This paper is comprised of four parts. Part 1 includes the abstract and the introduction, Part 2 describes the data file utilized throughout the paper as sample data, and final expected data. Part 3 show the codes step by step from source AE data to final expected AE data. Part 4 concludes the paper.

DATA FILE
A sample data file containing AE records is used in the SAS program for illustration purposes. The variables and records in the data file (raw.ae) are the following:

<table>
<thead>
<tr>
<th>SUBID</th>
<th>AEDEDOC</th>
<th>AESTRT</th>
<th>AENDRN</th>
<th>AERER</th>
<th>AESEX</th>
<th>AEREL</th>
<th>AEACR</th>
<th>AEOOT</th>
<th>AEORDN</th>
</tr>
</thead>
<tbody>
<tr>
<td>00008</td>
<td>NASOPHARYNGITIS</td>
<td>17MAY2012</td>
<td>25AUG2012</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>Non-collapsible</td>
</tr>
<tr>
<td>00010</td>
<td>ASTHMA</td>
<td>11JAN2012</td>
<td>18FEB2012</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>Type I: Same onset date</td>
</tr>
<tr>
<td>00017</td>
<td>MEFENDO</td>
<td>19JUL2012</td>
<td>19SEP2012</td>
<td>1</td>
<td>1</td>
<td>8</td>
<td>1</td>
<td>1</td>
<td>Type II: Contiguous sequence</td>
</tr>
<tr>
<td>00017</td>
<td>OTTIS MEDIA ACUTE</td>
<td>27MAY2013</td>
<td>27AUG2013</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>Type III: Overlapping</td>
</tr>
<tr>
<td>00017</td>
<td>OTTIS MEDIA ACUTE</td>
<td>27MAY2013</td>
<td>27AUG2013</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Final AE records are expected as follow:

<table>
<thead>
<tr>
<th>SUBID</th>
<th>AEDEDOC</th>
<th>AESTRT</th>
<th>AENDRN</th>
<th>AERER</th>
<th>AESEX</th>
<th>AEREL</th>
<th>AEACR</th>
<th>AEOOT</th>
<th>AEORDN</th>
</tr>
</thead>
<tbody>
<tr>
<td>00008</td>
<td>NASOPHARYNGITIS</td>
<td>17MAY2012</td>
<td>25AUG2012</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>Non-collapsible</td>
</tr>
<tr>
<td>00010</td>
<td>ASTHMA</td>
<td>11JAN2012</td>
<td>18FEB2012</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>Type I: Collapsed</td>
</tr>
<tr>
<td>00017</td>
<td>MEFENDO</td>
<td>19JUL2012</td>
<td>19SEP2012</td>
<td>2</td>
<td>1</td>
<td>8</td>
<td>2</td>
<td>2</td>
<td>Type II: Collapsed</td>
</tr>
<tr>
<td>00017</td>
<td>OTTIS MEDIA ACUTE</td>
<td>27MAY2013</td>
<td>27AUG2013</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>Type III: Collapsed</td>
</tr>
<tr>
<td>00035</td>
<td>MEFENDO</td>
<td>07JUL2012</td>
<td>22JUL2012</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>Non-collapsible</td>
</tr>
</tbody>
</table>

SAS CODES:
Step by step follow flow chart:

```
* macro for grouping Multiple/Non-Multiple AE records
%macro classifyAE(indsn=, outdsn=, grpvar=, whr=);
  proc sql;
    create table & outdsn. as
    select *
    from & indsn.
    group by & grpvar.
    & whr.
  ;quit;
%mend classifyAE;
```

```
* Non-Multiple AE records from source AE ---> Dataset: NonMultipleAE
%classifyAE(indsn=raw.ae, outdsn=NonMultipleAE, grpvar=%str(subjid, aedecod), whr=having count(*)=1);
```

```
* Multiple AE records from source AE ---> Dataset: MultipleAE
%classifyAE(indsn=raw.ae, outdsn=MultipleAE, grpvar=%str(subjid, aedecod), whr=having count(*)>1);
```

```
* Not Type I records from MultipleAE ---> Dataset: NonTypel
%classifyAE(indsn=MultipleAE, outdsn=NonTypel, grpvar=%str(subjid, aedecod, aestdn), whr=having count(*)=1);
```
* Type I: AE records have same onset date from MultipleAE, collapsed --> Dataset: CollapsedType1

```sql
proc sql;
create table CollapsedType1 as
    select distinct subjid, aedecod, aestdn, max(aeser) as aeser, max(aesev) as aesev,
        max(aerel) as aerel, max(aeacn) as aeacn, max(aeout) as aeout, max(aecontrt) as aecontrt,
            case when sum(aeendn=0) then .
        else max(aeendn)
    end as aeendn
    from MultipleAE
    group by subjid, aedecod, aestdn
    having count(*)>1
;quit;
```

* Set Not Type I records and collapsed Type I together --> Dataset: AfterColType1

```plaintext
data AfterColType1;
    set NonType1 CollapsedType1;
;run;
```

* Non-Multiple AE records from AfterColType1 --> Dataset: NonMultipleAEafter1

```plaintext
%classifyAE(indsn=AfterColType1, outdsn=NonMultipleAEafter1, grpvar=%str(subjid, aedecod), whr=having count(*)=1);
```

* Multiple AE records from AfterColType1 --> Dataset: MultipleAEafter1

```plaintext
%classifyAE(indsn=AfterColType1, outdsn=MultipleAEafter1, grpvar=%str(subjid, aedecod), whr=having count(*)>1);
```

* Type II: Contiguous AE records from MultipleAEafter1, collapsed -->Dataset: CollapsedType2

```plaintext
proc sort data=MultipleAEafter1 out=ConBase; by subjid aedecod aestdn aeendn; run;
```

```plaintext
data step1_ConAE;
    format aeendn_lag date9.;
    set ConBase;
        by subjid aedecod aestdn aeendn;
        aeendn_lag=lag(aeendn);
        if first.aedecod then aeendn_lag=.;
;run;
```

```plaintext
data step2_ConAE;
    set step1_ConAE;
    format aeendn_lag date9.;
    by subjid aedecod aestdn aeendn;
    if N(aestdn,aeendn_lag)=2 and (aestdn=aeendn_lag or aestdn=aeendn_lag+1) then ConFlag=1;
    if aeendn-. then OngoingFlag=1;
;run;
```

```plaintext
data step3_ConAE;
    merge step2_ConAE(rename=(ConFlag=in1_ConFlag))
        step2_ConAE(firstobs=2 keep=ConFlag rename=(ConFlag=in2_ConFlag));
```

* Type II: Contiguous AE records from MultipleAEafter1, collapsed -->Dataset: CollapsedType2

```plaintext
* onset date of an AE is either the same day or 1 day later than the resolution date of a preceding AE
* if one resolution date is missing, the collapsed event will be considered unresolved(ongoing)
if N(aestdn,aeendn_lag)=2 and (aestdn=aeendn_lag or aestdn=aeendn_lag+1) then ConFlag=1;
if aeendn-. then OngoingFlag=1;
```

```plaintext
run;
```

```plaintext
data step3_ConAE;
    merge step2_ConAE(rename=(ConFlag=in1_ConFlag))
        step2_ConAE(firstobs=2 keep=ConFlag rename=(ConFlag=in2_ConFlag));
```

* Since variable ConFlag is derived from last record, then last record should have ConFlag=1
* if ConFlag=1 in current record. Merge ConFlag by using condition firstobs=2 and without merge
* by variables to get this information.

```plaintext
run;
```
ConFlag=coalesce(in1_ConFlag,in2_ConFlag);
drop in1_ConFlag in2_ConFlag;

data notContiguous Contiguous;
set step3_ConAE;
if ConFlag^=1 then output notContiguous;
else output Contiguous;

run;

proc sql;
create table CollapsedType2 as
select distinct subjid, aedecod, max(aesev) as aesev, max(aeser) as aeser, max(aerel) as aerel, max(aeacn) as aeacn, max(aeout) as aeout, max(aecontrt) as aecontrt,
case when sum(OngoingFlag=1)>0 then .
else max(aeendn)
end as aeendn
, min(aestdn) as aestdn
from Contiguous
group by subjid, aedecod
;quit;

************************************************************************************
* Set Not Type II records and collapsed Type II together ---> Dataset: AfterColType2
************************************************************************************

data AfterColType2;
set CollapsedType2 notContiguous;

run;

*****************************************************************************
* Non-Multiple AE records from AfterColType2 ---> Dataset: NonMultipleAEafter2
*****************************************************************************;
%classifyAE(indsn=AfterColType2, outdsn=NonMultipleAEafter2, grpvar=%str(subjid, aedecod), whr=having count(*)=1);

*****************************************************************************
* Multiple AE records from AfterColType2 ---> Dataset: MultipleAEafter2
*****************************************************************************;
%classifyAE(indsn=AfterColType2, outdsn=MultipleAEafter2, grpvar=%str(subjid, aedecod), whr=having count(*)>1);

*****************************************************************************
* Type III: Overlap AE records from MultipleAEafter2,collapsed ---> Dataset: CollapsedType3
*****************************************************************************;
proc sort data=MultipleAEafter2 out=OverBase; by subjid aedecod aestdn aeendn; run;
data step1_OverAE; format aestdn_lag aeendn_lag date9.;
set OverBase;
by subjid aedecod aestdn aeendn;
aestdn_lag=lag(aestdn);
aeendn_lag=lag(aeendn);
if first.aedecod then do;
aestdn_lag=aestdn;
aeendn_lag=aeendn;
end;
run;
data step2_OverAE;
set step1_OverAE;
* onset date of an AE is after the onset date and prior to the resolution date of a preceding AE
* if one resolution date is missing, the collapsed event will be considered unresolved(ongoing)
if aestdn_lag>aestdn and (aestdn=aeendn_lag or aeendn_lag=. ) then OverFlag=1;
if aeendn=. then OngoingFlag=1;
run;
data step3_OverAE;
merge
step2_OverAE(rename=(OverFlag=in1_OverFlag))
step2_OverAE(firstobs=2 keep=OverFlag rename=(OverFlag=in2_OverFlag));
 ************************************************************************************
* Since variable ConFlag is derived from last record, then last record should have OverFlag=1
* if OverFlag=1 in current record. Merge ConFlag by using condition firstobs=2 and without
* merge by variables to get this information.
 ************************************************************************************;
OverFlag=coalesce(in1_OverFlag,in2_OverFlag);
drop in1_OverFlag in2_OverFlag;
run;
data notOverlap Overlap;
set step3_OverAE;
if OverFlag^=1 then output notOverlap;
else output Overlap;
run;
proc sql;
create table CollapsedType3 as
select distinct subjid, aedecod, max(aesev) as aesev, max(aeser) as aeser, max(aerel) as aerel,
max(aeacn) as aeacn, max(aeout) as aeout, max(aecontrt) as aecontrt,
case when sum(OngoingFlag=1)>0 then .
else max(aeendn)
end as aeendn,
min(aestdn) as aestdn
from Overlap
group by subjid, aedecod
;quit;
************************************************************************************
* Set Not Type III records and collapsed Type III together ---> Dataset: AfterColType3
*************************************************************************************;
data AfterColType3;
set CollapsedType3 notOverlap;
run;
*************************************************************************************
* Get final AE by setting:
* NonMultipleAE <- Non-Multiple AE records from source AE
* NonMultipleAEafter1 <- Non-Multiple AE records after collapse type I
* NonMultipleAEafter2 <- Non-Multiple AE records after collapse type II
* AfterColType3 <- Non-Overlap AE and collapsed type III records after collapse type II
************************************************************************************
data FinalAE;
set NonMultipleAE
NonMultipleAEafter1
NonMultipleAEafter2
AfterColType3;
drop aestdn_lag aeendn_lag ConFlag OverFlag OngoingFlag
;run;

Some other AE variables maybe also needed base on specify condition. In this case, we need to replace "distinct
subjid, aedecod" to "*", and update the codes "max(aexxx) as aexxx" to another temp variable in each SQL
procedure which been used for collapsing, and then add more procedure step and data step(e.g., sort by variables,
and then use first/last by variables to select the needed record) to get final AE data.

CONCLUSION
This paper takes a comprehensive approach in conjunction with SAS data steps to perform AE records
collapsing. It provide sample SAS codes step by step to approach, contain collapse method for each type of
collapsible AE records. The collapsing algorithm implements worst case scenario.
REFERENCES
Collapsing Adverse Experiences Records, Shi-Tao Yeh, EDP Contract Services, Bala Cynwyd, PA

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

   Name: Haiqiang Luo
   Enterprise: PPD Inc.
   Address: 8/F, Tower B, Central Point Plaza No 11, Dongzhimen South Ave. Dongcheng District
   City, State ZIP: Beijing, 100007
   Work Phone: +86 10-57636346
   Fax: +86 10-57636251
   E-mail: Haiqiang.Luo@ppdi.com
   Web: www.ppdi.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.

Other brand and product names are trademarks of their respective companies.