Paper DH10

Detecting Treatment Emergent Adverse Events (TEAEs)

Matthias Lehrkamp, Bayer AG, Berlin, Germany

ABSTRACT

Treatment emergent adverse event (TEAE) tables are mandatory in each clinical trial summary. An adverse event (AE) is counted as treatment emergent in any case when it starts or gets worse during a treatment period (ICH E9). At first glance, it looks simple to detect AEs as treatment emergent. Unfortunately, clinical data challenges you to fight with different input accuracy, e.g. the first treatment start date often includes time information whereas all other treatment dates simply present a date without time information. You have to react on partial and complete missing dates and some special cases like related previous AEs, AEs resulting in death, or patients where all dates are completely missing. Many exceptions and differentiations make this topic cumbersome. This paper will look at a method to detect TEAEs which works with all data situations.

INTRODUCTION

To detect TEAEs with partial and completely missing data or overlapping treatment exposure intervals is somehow impossible. The examples used in this paper are mainly based on the content of two other papers. A more general instruction about TEAEs is given in "*TEAE, Did I flag it right?*" published by Arun Raj Vidhyadharan and Sunil Mohan Jairath in 2015. The second paper was published in 2017 by Mengya Gillian Yin and Wen Tan, and focuses on crossover designs with the title "*How to define Treatment Emergent Adverse Event (TEAE) in crossover clinical trials?*". Both papers are not showing any SAS® code to consider the described data saturations. Therefore, this paper shows a straightforward algorithm to handle partial and complete missing data or overlapping treatment exposure intervals, and works properly in all listed cases of both papers. In addition, the algorithm presented is able to cope with longer interruptions within a treatment period.

HOW TO START

The overall definition of a TEAE is given in the GCP ICH E9 guideline and is considered in all cases:

"*An event that emerges during treatment having been absent pre-treatment, or worsens relative to the pre-treatment state.*"

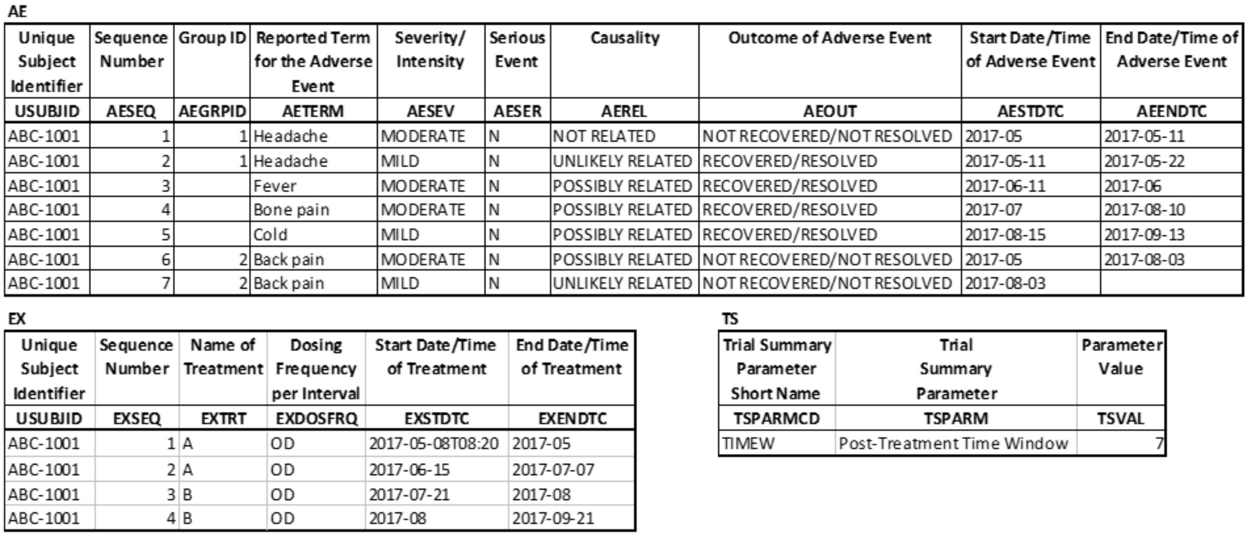
The definition is precise enough but it also includes wordings which can vary from study to study. The first question is about the understanding of "*during treatment*" in the study? This must be clearly defined in the clinical study protocol or in the Statistical Analysis Plan (SAP), respectively. Usually the "*during treatment*" period is the period which starts with the first treatment exposure and stops after a predefined post-treatment time window. The post-treatment time window considers the time to clear the body from the drug and depends on the pharmacological characteristics of the substance studied. Both, the exposure times and the post-treatment time window are required as input to this algorithm. The next section gives an overview of where the input data comes from and which data is used in this paper.

WHERE TO FIND THE DATA

All examples and rules are based on SDTM data as input data. If a different data model is used, translations may be necessary. A good reason to use the SDTM model is the fixed structure and the good description of the SDTM metadata. Although the FDA requires a TEAE flag in SDTM, this paper concentrates on building the TEAF for the analysis dataset ADAE. This is useful, as the analysis TEAE flag can have several additions, as well as derived information can be stored to support the TEAE analysis. However, this paper can also be used to create the overall TEAE flag in SDTM. It is also a good practice to check the consistency between the TEAE flags in SDTM and ADaM.

The domains AE and SUPPAE contain information about adverse events. The domain AE contains the main information, while the domain SUPPAE contains the TEAE flag and some additional non-standardized data. The post-treatment time window should be stored in the trial summary domain TS. The information about the treatment exposure comes from the domain EX. It depends on the type of exposure, whether individual exposure times or intervals are present. Examples of these datasets are given in Figure 1.

Figure 1: Example for the required SDTM data



AE

Unique Subject Identifier	Sequence Number	Group ID	Reported Term for the Adverse Event	Severity/ Intensity	Serious Event	Causality	Outcome of Adverse Event	Start Date/Time of Adverse Event	End Date/Time of Adverse Event
USUBJID	AESEQ	AEGRPID	AETERM	AESEV	AESER	AEREL	AEOUT	AESTDTC	AEENDTC
ABC-1001	1	1	Headache	MODERATE	N	NOT RELATED	NOT RECOVERED/NOT RESOLVED	2017-05	2017-05-11
ABC-1001	2	1	Headache	MILD	N	UNLIKELY RELATED	RECOVERED/RESOLVED	2017-05-11	2017-05-22
ABC-1001	3		Fever	MODERATE	N	POSSIBLY RELATED	RECOVERED/RESOLVED	2017-06-11	2017-06
ABC-1001	4		Bone pain	MODERATE	N	POSSIBLY RELATED	RECOVERED/RESOLVED	2017-07	2017-08-10
ABC-1001	5		Cold	MILD	N	POSSIBLY RELATED	RECOVERED/RESOLVED	2017-08-15	2017-09-13
ABC-1001	6	2	Back pain	MODERATE	N	POSSIBLY RELATED	NOT RECOVERED/NOT RESOLVED	2017-05	2017-08-03
ABC-1001	7	2	Back pain	MILD	N	UNLIKELY RELATED	NOT RECOVERED/NOT RESOLVED	2017-08-03	

EX

Unique Subject Identifier	Sequence Number	Name of Treatment	Dosing Frequency per Interval	Start Date/Time of Treatment	End Date/Time of Treatment
USUBJID	EXSEQ	EXTRT	EXDOSFRQ	EXSTDTC	EXENDTC
ABC-1001	1	A	OD	2017-05-08T08:20	2017-05
ABC-1001	2	A	OD	2017-06-15	2017-07-07
ABC-1001	3	B	OD	2017-07-21	2017-08
ABC-1001	4	B	OD	2017-08	2017-09-21

TS

Trial Summary Parameter Short Name	Trial Summary Parameter	Parameter Value
TSPARMCD	TSPARM	TSVAL
TIMEW	Post-Treatment Time Window	7

The exposure dataset used as input for this algorithm includes only the treatment exposure taken by subject. As an alternative to the domain EX, it is possible to use the subject elements domain SE. This dataset includes the epochs of each subject. A disadvantage is that longer interruptions in a treatment epoch cannot be recognized. Therefore, the recommendation is to work with the domain EX.

THE INTERVAL BASED ALGORITHM (IBA)

The main part of the algorithm is to generate an interval for the AE start and to compare this with the exposure intervals. The AE start interval consists of the earliest and latest possible starting time. The occurrence of an event during the treatment is given when the AE start interval has a match with any exposure interval. As two intervals are compared, this algorithm is called Interval Based Algorithm (IBA). The advantage is that the AE date/time can be determined independently of the exposure data. This results in a stable, straightforward algorithm that applies to all data cases. The only exceptions are data issues like when the AE start date is after the AE end date. Therefore, it is also a good practice to carefully check the data and to share data issues with the study team.

The following paragraphs will guide the reader through the necessary steps following an IBA approach.

Step 1: Build intervals of all date/times in which at least the year is present

In the first step, all date/times are calculated up to the precision of seconds. This step is limited to all date/times in which at least the year is present. Each date results in an interval, beginnings with the earliest possible time (*DT_MIN) and ends with the last possible time (*DT_MAX). In case the start/date time is complete, the start and end time of the interval is the same. The replacement of missing date and time parts is as shown in Table 1.

Table 1: Replacements of missing date and time parts

	Earliest date/time	Last date/time
Month	1	12
Day	1	Last day of the month in the given year
Hour	0	23
Minute	0	59
Second	0	59

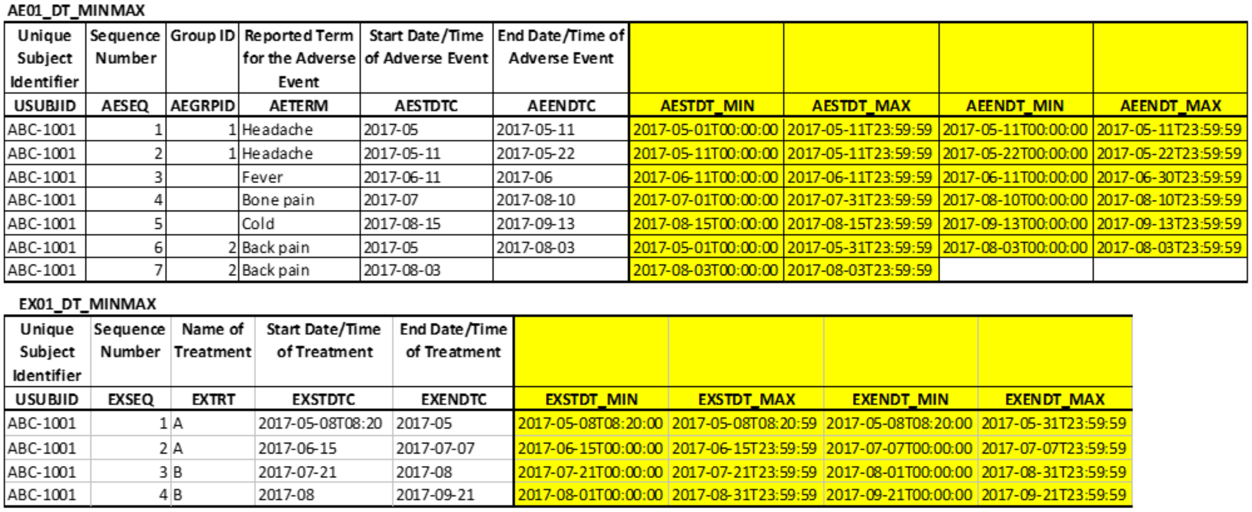
All date/time parts should be stored as numerical value using the date/time format IS8601DT. After the imputation of missing parts with the replacements specified in Table 1, the date/time values must be checked for consistency and must be adapted if necessary. The last possible start date/time must be limit by the last possible end date/time, as well as the earliest possible end date/time must be limit by the earliest possible start date/time.

The following data step shows the most important SAS statements to replace the AE start date/time, where at least the year is present. The AE end date/time should be replaced similarly in the same data step. A placeholder shows where the program code must be inserted.

```
/* impute the AE date/time */
DATA ae01_dt_minmax;
  SET sdtm.ae;
  * set regular expression ID to parse is8601dt formatted date/time parts;
  prxid= prxparse(
    '/^(\d{4}|-)(-(\d{2}|-)(-(\d{2}|-)(T(\d{2}|-)(:(\d{2}|-)(:(\d{2}))?)?)?)?)? *$/');
  *** impute the start date/time ***;
  FORMAT aestdt_min aestdt_max is8601dt.;
  * check that date/time is not missing;
  IF NOT missing(aestdtc) THEN DO;
    * check that at least the year is present;
    IF prxmatch("/^\d{4}/",aestdtc)=0 THEN DO;
      PUT 'INFO: Date without year information is interpreted as missing. ' _ALL_;
    END;
    ELSE DO;
      * extract each date/time part;
      IF prxmatch(prxid,aestdtc) THEN DO;
        aestdt_year= input(prxposn(prxid,1,aestdtc), 4.);
        aestdt_min_month= input(prxposn(prxid,3,aestdtc), 2.);
        aestdt_min_day= input(prxposn(prxid,5,aestdtc), 2.);
        aestdt_min_hour= input(prxposn(prxid,7,aestdtc), 2.);
        aestdt_min_minute= input(prxposn(prxid,9,aestdtc), 2.);
        aestdt_min_second= input(prxposn(prxid,11,aestdtc), 2.);
      END;
      * impute month when missing;
      IF missing(aestdt_min_month) THEN DO;
        aestdt_min_month= 1;
        aestdt_max_month= 12;
      END;
      * if the month is not missing then the max month is the same as the min month;
      ELSE DO;
        aestdt_max_month= aestdt_min_month;
      END;
      * impute day when missing;
      IF missing(aestdt_min_day) THEN DO;
        aestdt_min_day= 1;
        * Last day of the month in the given year;
        aestdt_max_day= day(intnx('month',input(cats(put(aestdt_year,z4.)
                           ,'-',put(aestdt_max_month,z2.),'-01'),is8601da.),0,'E'));
      END;
      ELSE DO;
        aestdt_max_day= aestdt_min_day;
      END;
      * impute hour, minute and second similar to the month;
      * <<< add code here >>>;

      * create the full date/time value;
      aestdt_min= dhms(mdy(aestdt_min_month, aestdt_min_day, aestdt_year)
                      ,aestdt_min_hour,aestdt_min_minute,aestdt_min_second);
      aestdt_max= dhms(mdy(aestdt_max_month, aestdt_max_day, aestdt_year)
                      ,aestdt_max_hour,aestdt_max_minute,aestdt_max_second);
    END;
  END;
  *** impute the end date/time ***;
  * <<< repeat the code to impute the AE end date/time here >>>;

  *** check that the dates are consistent ***;
  IF nmiss(aestdt_min,aeendt_max)=0 AND aestdt_min > aeendt_max THEN DO;
    PUT 'ERROR: The earliest AE start is after the latest AE end. '
               'Both dates will be set to missing! ' _ALL_;
    CALL missing(aestdt_min,aestdt_max,aeendt_min,aeendt_max);
  END;
  * the last AE start date/time is limit to the last AE end date/time;
  IF n(aestdt_max,aeendt_max)=2 AND aestdt_max>aeendt_max THEN aestdt_max= aeendt_max;
  * the earliest AE end date/time is limit to the earliest AE start date/time;
  IF n(aeendt_min,aestdt_min)=2 AND aeendt_min<aestdt_min THEN aeendt_min= aestdt_min;
  KEEP usubjid aeseq aegrpid aeterm aesev aeser aerel aeout aestdtc aeendtc
       aestdt_min aestdt_max aeendt_min aeendt_max;
RUN;
```

Use a similar data step to impute the exposure date/times. Figure 2 shows the imputation of all date/times in which at least the year is present.

Figure 2: Imputation of date/times in which at least the year is present



AE01_DT_MINMAX

Unique Subject Identifier	Sequence Number	Group ID	Reported Term for the Adverse Event	Start Date/Time of Adverse Event	End Date/Time of Adverse Event				
USUBJID	AESEQ	AEGRPID	AETERM	AESTDTC	AEENDTC	AESTDT_MIN	AESTDT_MAX	AEENDT_MIN	AEENDT_MAX
ABC-1001	1	1	Headache	2017-05	2017-05-11	2017-05-01T00:00:00	2017-05-11T23:59:59	2017-05-11T00:00:00	2017-05-11T23:59:59
ABC-1001	2	1	Headache	2017-05-11	2017-05-22	2017-05-11T00:00:00	2017-05-11T23:59:59	2017-05-22T00:00:00	2017-05-22T23:59:59
ABC-1001	3		Fever	2017-06-11	2017-06	2017-06-11T00:00:00	2017-06-11T23:59:59	2017-06-11T00:00:00	2017-06-30T23:59:59
ABC-1001	4		Bone pain	2017-07	2017-08-10	2017-07-01T00:00:00	2017-07-31T23:59:59	2017-08-10T00:00:00	2017-08-10T23:59:59
ABC-1001	5		Cold	2017-08-15	2017-09-13	2017-08-15T00:00:00	2017-08-15T23:59:59	2017-09-13T00:00:00	2017-09-13T23:59:59
ABC-1001	6	2	Back pain	2017-05	2017-08-03	2017-05-01T00:00:00	2017-05-31T23:59:59	2017-08-03T00:00:00	2017-08-03T23:59:59
ABC-1001	7	2	Back pain	2017-08-03		2017-08-03T00:00:00	2017-08-03T23:59:59		

EX01_DT_MINMAX

Unique Subject Identifier	Sequence Number	Name of Treatment	Start Date/Time of Treatment	End Date/Time of Treatment				
USUBJID	EXSEQ	EXTRT	EXSTDTC	EXENDTC	EXSTDT_MIN	EXSTDT_MAX	EXENDT_MIN	EXENDT_MAX
ABC-1001	1	A	2017-05-08T08:20	2017-05	2017-05-08T08:20:00	2017-05-08T08:20:59	2017-05-08T08:20:00	2017-05-31T23:59:59
ABC-1001	2	A	2017-06-15	2017-07-07	2017-06-15T00:00:00	2017-06-15T23:59:59	2017-07-07T00:00:00	2017-07-07T23:59:59
ABC-1001	3	B	2017-07-21	2017-08	2017-07-21T00:00:00	2017-07-21T23:59:59	2017-08-01T00:00:00	2017-08-31T23:59:59
ABC-1001	4	B	2017-08	2017-09-21	2017-08-01T00:00:00	2017-08-31T23:59:59	2017-09-21T00:00:00	2017-09-21T23:59:59

The variables AEENDT_MIN, EXSTDT_MAX and EXENDT_MIN are not required for the IBA, but are listed for the completeness of the intervals.

Step 2: Impute all date/times where at least the year is missing

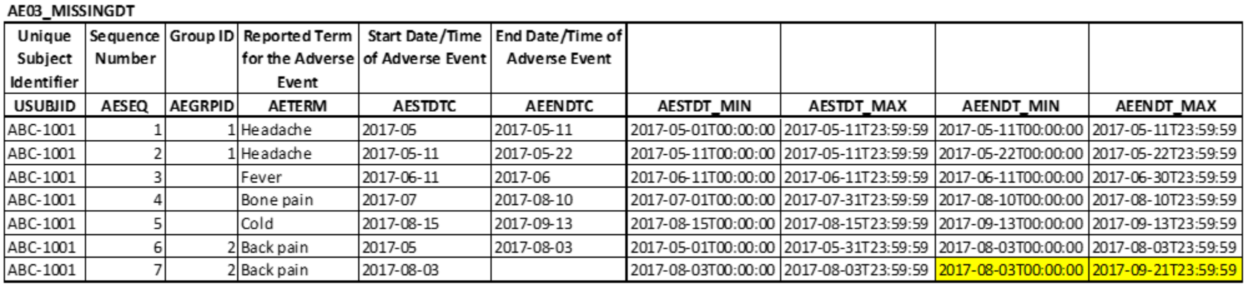
All date/times for which the year does not exist is considered to be completely missing. A meaningful replacement of the year is not possible. These date/times together with the missing data is replaced by the data from the 1st step. More specifically, a missing value results in an interval beginning with the earliest possible date/time and ending with the last possible date/time of the imputed values from the first step. In both situations, all imputed start and end date/times from both datasets are considered. As new data is added, they must be rechecked for consistency. The last possible start date/time will be limited by the last possible end date/time, as well as the earliest possible end date/time will be limited by the earliest possible start date/time. During this step, the post-treatment time window should also be added to the exposure data.

Subjects who do not have a date/time with at least the year information must be handled separately. These subjects are really special and should exist in unclean data. However, all these subjects are excluded from further TEAE detection in this step. In a later step, all AEs of these subjects are classified as treatment emergent.

```
/* Get the earliest and last possible imputed date/time */
PROC SQL;
  CREATE TABLE teae01_alldt AS
      SELECT usubjid, aestdt_min AS teaedt
        FROM ae01_dt_minmax WHERE NOT missing(aestdt_min)
UNION SELECT usubjid, aestdt_max AS teaedt
        FROM ae01_dt_minmax WHERE NOT missing(aestdt_max)
UNION SELECT usubjid, aeendt_min AS teaedt
        FROM ae01_dt_minmax WHERE NOT missing(aeendt_min)
UNION SELECT usubjid, aeendt_max AS teaedt
        FROM ae01_dt_minmax WHERE NOT missing(aeendt_max)
UNION SELECT usubjid, exstdt_min AS teaedt
        FROM ex01_dt_minmax WHERE NOT missing(exstdt_min)
UNION SELECT usubjid, exstdt_max AS teaedt
        FROM ex01_dt_minmax WHERE NOT missing(exstdt_max)
UNION SELECT usubjid, exendt_min AS teaedt
        FROM ex01_dt_minmax WHERE NOT missing(exendt_min)
UNION SELECT usubjid, exendt_max AS teaedt
        FROM ex01_dt_minmax WHERE NOT missing(exendt_max);
  CREATE TABLE teae02_dt_minmax AS
    SELECT usubjid
      , min(teaedt) AS teaedt_min FORMAT=is8601dt. LABEL='Earliest Possible Date/Time'
      , max(teaedt) AS teaedt_max FORMAT=is8601dt. LABEL='Last Possible Date/Time'
      FROM teae01_alldt
      GROUP BY usubjid;
QUIT;
```

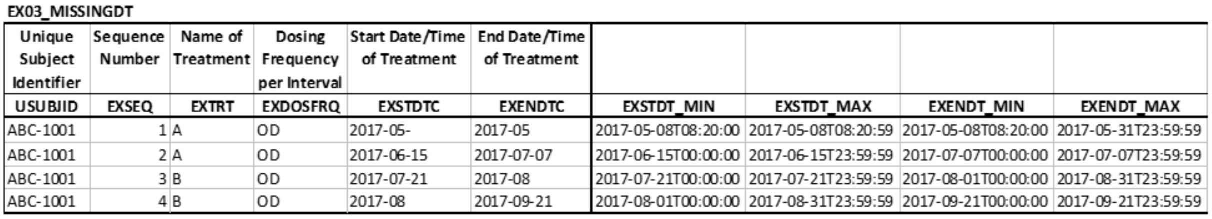
```
/* Join the min/max date/times and drop subjects with only missing dates */
PROC SQL;
  CREATE TABLE ae02_teaedt AS
    SELECT l.*, r.teaedt_min, r.teaedt_max
      FROM ae01_dt_minmax AS l
      /* The inner join drops subjects with only missing dates */
      INNER JOIN teae02_dt_minmax AS r
        ON l.usubjid = r.usubjid
  ;
  CREATE TABLE ex02_teaedt AS
    SELECT l.*, r.teaedt_min, r.teaedt_max
         , INPUT(r2.tsval,best.) AS timew LABEL="Post Treatment Time Window (days)"
      FROM ex01_dt_minmax AS l
      INNER JOIN teae02_dt_minmax AS r
        ON l.usubjid = r.usubjid
      /* Add the post-treatment time window */
      LEFT JOIN sdtm.ts( WHERE= ( tsparmcd='TIMEW' ) ) AS r2
        ON 1
  ;
QUIT;

/* Replace missing AE date/times with the absolute min/max date/times */
DATA ae03_missingdt;
  SET ae02_teaedt;
  IF missing(aestdt_min) THEN DO;
    aestdt_min= teaedt_min;
    aestdt_max= teaedt_max;
  END;
  IF missing(aeendt_min) THEN DO;
    aeendt_min= teaedt_min;
    aeendt_max= teaedt_max;
  END;
  * the last AE start date/time is limit to the last AE end date/time;
  IF n(aestdt_max,aeendt_max)=2 AND aestdt_max > aeendt_max
    THEN aestdt_max= aeendt_max;
  * the earliest AE end date/time is limit to the earliest AE start date/time;
  IF n(aeendt_min,aestdt_min)=2 AND aeendt_min < aestdt_min
    THEN aeendt_min= aestdt_min;
RUN;

/* Replace missing exposure date/times with the absolute min/max date/times */
DATA ex03_missingdt;
  SET ex02_teaedt;
  IF missing(exstdt_min) THEN DO;
    exstdt_min= teaedt_min;
    exstdt_max= teaedt_max;
  END;
  IF missing(exendt_min) THEN DO;
    exendt_min= teaedt_min;
    exendt_max= teaedt_max;
  END;
  * the last ex start date/time is limit to the last ex end date/time;
  IF n(exstdt_max,exendt_max)=2 AND exstdt_max > exendt_max
    THEN exstdt_max= exendt_max;
  * the earliest ex end date/time is limit to the earliest ex start date/time;
  IF n(exendt_min,exstdt_min)=2 AND exendt_min < exstdt_min
    THEN exendt_min= exstdt_min;
RUN;
```

Figure 3: Imputation of date/times where at least the year is not present



AE03_MISSINGDT

Unique Subject Identifier	Sequence Number	Group ID	Reported Term for the Adverse Event	Start Date/Time of Adverse Event	End Date/Time of Adverse Event				
USUBJID	AESEQ	AEGRPID	AETERM	AESTDTC	AEENDTC	AESTDT_MIN	AESTDT_MAX	AEENDT_MIN	AEENDT_MAX
ABC-1001	1	1	Headache	2017-05	2017-05-11	2017-05-01T00:00:00	2017-05-11T23:59:59	2017-05-11T00:00:00	2017-05-11T23:59:59
ABC-1001	2	1	Headache	2017-05-11	2017-05-22	2017-05-11T00:00:00	2017-05-11T23:59:59	2017-05-22T00:00:00	2017-05-22T23:59:59
ABC-1001	3		Fever	2017-06-11	2017-06	2017-06-11T00:00:00	2017-06-11T23:59:59	2017-06-11T00:00:00	2017-06-30T23:59:59
ABC-1001	4		Bone pain	2017-07	2017-08-10	2017-07-01T00:00:00	2017-07-31T23:59:59	2017-08-10T00:00:00	2017-08-10T23:59:59
ABC-1001	5		Cold	2017-08-15	2017-09-13	2017-08-15T00:00:00	2017-08-15T23:59:59	2017-09-13T00:00:00	2017-09-13T23:59:59
ABC-1001	6	2	Back pain	2017-05	2017-08-03	2017-05-01T00:00:00	2017-05-31T23:59:59	2017-08-03T00:00:00	2017-08-03T23:59:59
ABC-1001	7	2	Back pain	2017-08-03		2017-08-03T00:00:00	2017-08-03T23:59:59	2017-08-03T00:00:00	2017-09-21T23:59:59



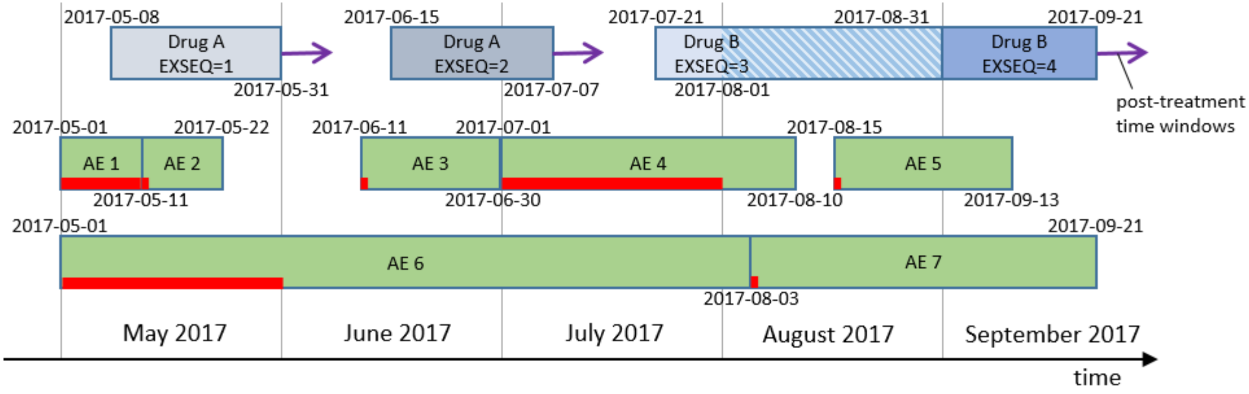
EX03_MISSINGDT

Unique Subject Identifier	Sequence Number	Name of Treatment	Dosing Frequency per Interval	Start Date/Time of Treatment	End Date/Time of Treatment				
USUBJID	EXSEQ	EXTRT	EXDOSFRQ	EXSTDTC	EXENDTC	EXSTDT_MIN	EXSTDT_MAX	EXENDT_MIN	EXENDT_MAX
ABC-1001	1	A	OD	2017-05-	2017-05	2017-05-08T08:20:00	2017-05-08T08:20:59	2017-05-08T08:20:00	2017-05-31T23:59:59
ABC-1001	2	A	OD	2017-06-15	2017-07-07	2017-06-15T00:00:00	2017-06-15T23:59:59	2017-07-07T00:00:00	2017-07-07T23:59:59
ABC-1001	3	B	OD	2017-07-21	2017-08	2017-07-21T00:00:00	2017-07-21T23:59:59	2017-08-01T00:00:00	2017-08-31T23:59:59
ABC-1001	4	B	OD	2017-08	2017-09-21	2017-08-01T00:00:00	2017-08-31T23:59:59	2017-09-21T00:00:00	2017-09-21T23:59:59

Step 3: Join the AE and the exposure data and determine treatment emergent for each single AE

Figure 4 shows a time diagram of the exposure intervals and the AE intervals as boxes for the imputed data from Figure 3. The boxes themselves are starting with the earliest imputed start date/time (*STDT_MIN) and ends with the last possible end date/time (*ENDT_MAX). The red bar in the AE box represents the AE start interval (AESTDT_MIN to AESTDT_MAX). If an AE start interval (red bar) overlaps with any exposure interval (blue and grey boxes) then the respected AE is treatment emergent.

Figure 4: Timeline of imputed data



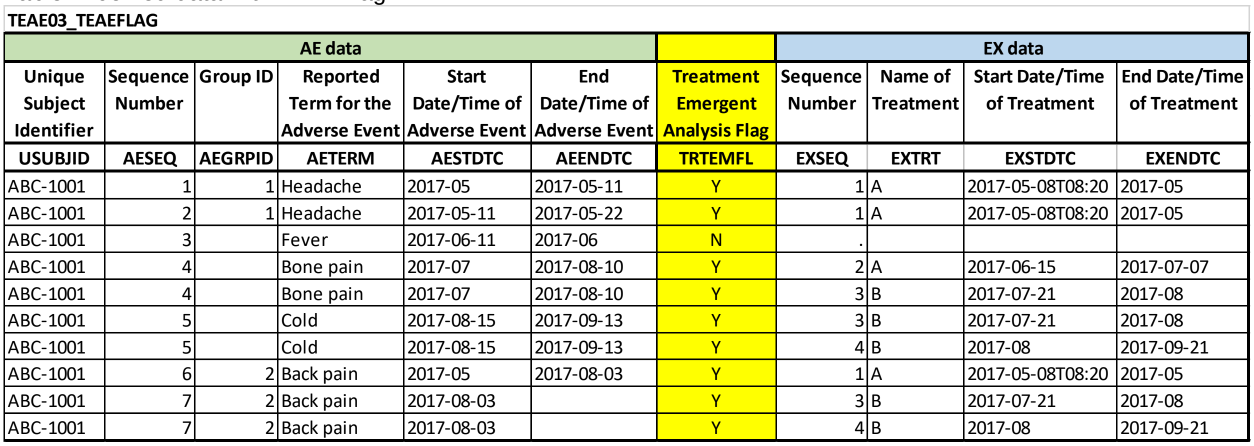
All date/times are imputed and ready for the merge. Since two intervals are compared, two conditions must be fulfilled in order to establish an overlap in time of the two intervals. The beginning of the first interval must be before or equal to the end of the second interval and the end of the first interval must be later or equal to the start of the second interval. An AE is counted as treatment emergent when the AE start interval overlaps at least one exposure interval. Note that the end of the exposure interval must be extended by the post-treatment time window in seconds. Therefore, the days are converted to seconds by calculating the time window in days * 24 * 60 * 60.

```
aestdt_min <= exendt_max + timew*24*60*60 AND aestdt_max >= exstdt_min
```

In all cases where intervals are matching, the TEAE flag (TRTEMFL) must be set to Y (=yes), otherwise it must be set to N (=no). Please note that currently most of the studies define a treatment emergency period from the first treatment exposure to the last treatment exposure plus the post treatment time windows. In that case the SE dataset can be used as input or the exposure dataset must be adapted accordingly.

```
PROC SQL;
  CREATE TABLE teae03_teaeflag AS
           /* AE data variables */
    SELECT l.usubjid, l.aeseq, l.aegrpid, l.aeterm, l.aesev, l.aeser, l.aerel
         , l.aeout, l.aestdtc, l.aeendtc, l.aestdt_min, l.aestdt_max, l.aeendt_min
         , l.aeendt_max
           /* Treatment emergent flag */
         , CASE WHEN NOT missing(r.usubjid) THEN 'Y'
                                            ELSE 'N'
           END AS trtemfl LABEL='Treatment Emergent Analysis Flag'
           /* variables from exposure data */
         , r.exseq, r.extrt, r.exstdtc, r.exendtc, r.exstdt_min, r.exstdt_max
         , r.exendt_min, r.exendt_max
      FROM ae03_missingdt AS l
      LEFT JOIN ex03_missingdt AS r
        ON     l.usubjid = r.usubjid
               /* the earliest AE start must be prior or equal to the latest treatment
                  exposure end date/time + post-treatment time window */
           AND aestdt_min <= exendt_max + timew * 24 * 60 * 60
               /* the latest AE start date must be after or equal to the earliest
                  treatment exposure start date/time */
           AND aestdt_max >= exstdt_min
      ORDER BY aeseq, exseq
  ;
QUIT;
```

Table 2: Joined data with TEAE flag

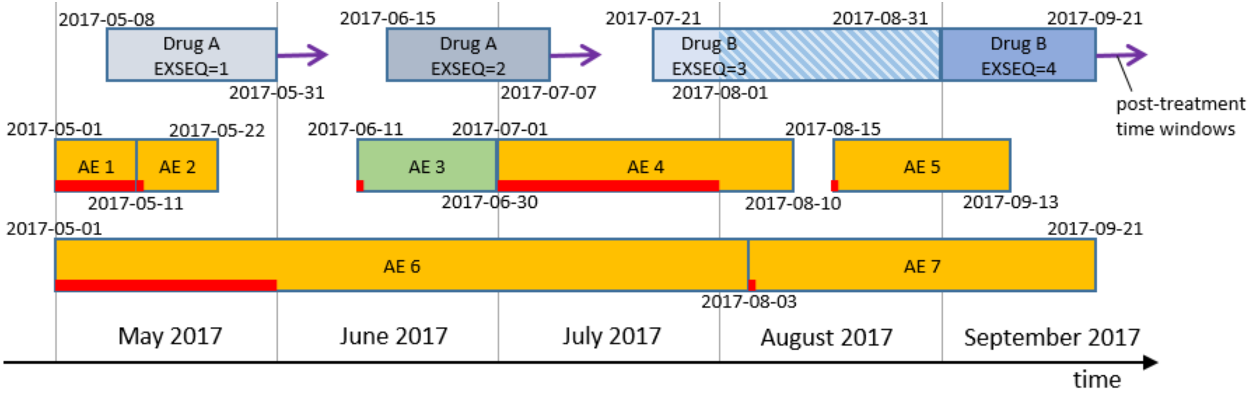


TEAE03_TEAEFLAG

AE data							EX data				
Unique Subject Identifier	Sequence Number	Group ID	Reported Term for the Adverse Event	Start Date/Time of Adverse Event	End Date/Time of Adverse Event	Treatment Emergent Analysis Flag	Sequence Number	Name of Treatment	Start Date/Time of Treatment	End Date/Time of Treatment
USUBJID	AESEQ	AEGRPID	AETERM	AESTDTC	AEENDTC	TRTEMFL	EXSEQ	EXTRT	EXSTDTC	EXENDTC
ABC-1001	1	1	Headache	2017-05	2017-05-11	Y	1	A	2017-05-08T08:20	2017-05
ABC-1001	2	1	Headache	2017-05-11	2017-05-22	Y	1	A	2017-05-08T08:20	2017-05
ABC-1001	3		Fever	2017-06-11	2017-06	N	.			
ABC-1001	4		Bone pain	2017-07	2017-08-10	Y	2	A	2017-06-15	2017-07-07
ABC-1001	4		Bone pain	2017-07	2017-08-10	Y	3	B	2017-07-21	2017-08
ABC-1001	5		Cold	2017-08-15	2017-09-13	Y	3	B	2017-07-21	2017-08
ABC-1001	5		Cold	2017-08-15	2017-09-13	Y	4	B	2017-08	2017-09-21
ABC-1001	6	2	Back pain	2017-05	2017-08-03	Y	1	A	2017-05-08T08:20	2017-05
ABC-1001	7	2	Back pain	2017-08-03		Y	3	B	2017-07-21	2017-08
ABC-1001	7	2	Back pain	2017-08-03		Y	4	B	2017-08	2017-09-21

The updated timeline diagram (Figure 5) shows all TEAEs filled in orange. It must be noted that the grouping of the AEs has not yet been considered up to this step. This will be the task of the next step.

Figure 5: Timeline of imputed data with TEAE flag



Step 4: Clean up duplicate AE records within the same treatment epoch

After merging the data, the same AE may be present in multiple lines (see AESEQ in 4, 5, and 7 in Table 2). This is possible when several exposure observations have a match with the same AE start interval, e.g. AE#4 vs. Drug A and Drug B. AE records belonging to different treatments must be kept, while observations with the same treatment need to be cleaned. In case the treatment is the same in two observations, the exposure interval with the closest previous start date/time (EXSTDT_MIN) to the earliest possible AE start (AESTDT_MIN) must be selected. If there are no previous exposure intervals with a start date before the AE start, the next interval starting after the AE start is selected. In case of duplicate exposure records, the sequence number is taken to select a specific observation.

```
/* Create variable pretrtfl to flag observations before the AE start.
   This information is used to select the closest exposure record to the AE start. */
DATA teae04_preaefl;
  SET teae03_teaeflag;
  ATTRIB pretrtfl LENGTH=$1 LABEL='Exposure is before AE Start Flag';
  IF exstdt_min < aestdt_min THEN pretrtfl= "Y";
                             ELSE pretrtfl= "N";
RUN;
/* AESEQ is used to select a specific observation by duplicate exposure records. */
PROC SORT DATA=teae04_preaefl OUT=teae05_sorted;
  BY usubjid aeseq extrt pretrtfl exstdt_min exseq;
RUN;
/* Selection of one exposure recored by aeseq and extrt */
DATA teae06_cleaned;
  SET teae05_sorted;
  BY usubjid aeseq extrt pretrtfl exstdt_min exseq;
  * indicator if a record is already selected;
  RETAIN teaeselectedfl;
  LENGTH teaeselectedfl $1;
  IF FIRST.extrt THEN teaeselectedfl='N';
  /* select the last exposure interval which starts before AE start, if available */
  IF teaeselectedfl='N' AND pretrtfl='Y' AND LAST.pretrtfl THEN DO;
    teaeselectedfl= 'Y';
    OUTPUT;
  END;
  /* select the first exposure interval next after the AE start, if available */
  ELSE IF teaeselectedfl='N' AND pretrtfl='N' AND FIRST.pretrtfl THEN DO;
    teaeselectedfl= 'Y';
    OUTPUT;
  END;
RUN;
```

In the given example both records for bone pain (AESEQ=4) must be kept as different treatments are present. The AE term cold (AESEQ = 5) is associated with the exposure observation EXSEQ=4 since it is the closest start date/time before the AE start. The same applies to the back pain (AESEQ=7). Summarized the observations with AESEQ=5, EXSEQ=3 and AESEQ=7, EXSEQ=3 are deleted in this step.

Step 5: Check grouped AEs and correct the TEAE flag for AEs that have not worsened

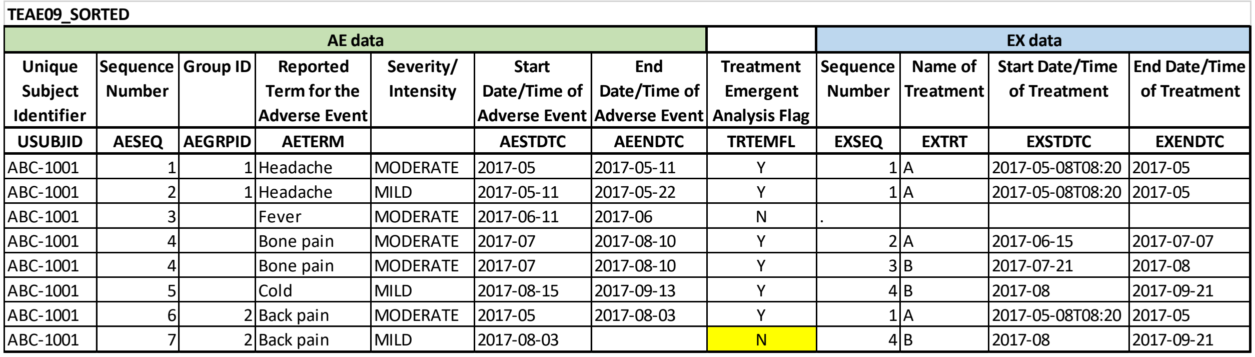
Now all AE observations are handled as single AEs. Due to the definition that only new AEs or AEs that have worsened are counted as TEAEs, an additional step is required to modify the TEAE flag for “grouped” AEs. A worsening of an AE is given when a worsening in the intensity of the AE is present or when a change from non-severe AE to severe AE occurred. The investigator is usually instructed to only add a new record of the same AE when the AE is worsening. Therefore, AE#2 and AE#7 are more theoretical. In cases in which an AE persists, while a new treatment medication started, the influence of the new medication is completely unclear. Those parts of grouped AEs which are not worsened but occur in another treatment will be set to non-treatment emergent (TEAE flag is set to N).

```
/* sort data for the next step */
PROC SORT DATA=teae06_cleaned OUT=teae07_cleaned;
    BY usubjid aegrpid aeterm aestdt_min aesev aeser;
RUN;
```

```
/* Use group information given in a variable */
DATA teae08_improvedaes;
  SET teae07_cleaned;
  BY usubjid;
  * Variable to save the values of the last record;
  LENGTH last_aeterm last_aegrpid last_aeout last_aeser last_aesev last_aerel
         last_aeout last_trtemfl last_extrt $200;
  RETAIN last_aeterm last_aegrpid last_aeout last_aeser last_aesev last_aerel
         last_aeout last_trtemfl last_extrt;
  * reset group variable for each subject;
  IF FIRST.usubjid THEN last_aegrpid= '';
  * reset all variables when aegrpid is missing or has a new ID;
  IF missing(aegrpid) OR aegrpid~=last_aegrpid THEN DO;
    CALL missing(last_aeterm, last_aeout, last_aeser, last_aesev, last_trtemfl,
                 last_extrt);
  END;
  * previous AE belongs to the current AE - check if TEAE flag must be changed;
  ELSE DO;
    * Check that grouped AE have the same AETERM;
    IF last_aeterm ~= aeterm
      THEN PUT 'WARNING: Found grouped AEs with different AE terms! ' _ALL_;
    * Check if AEOUT condition is fulfilled;
    IF last_aeout NOT IN('RECOVERING/RESOLVING' 'NOT RECOVERED/NOT RESOLVED')
      THEN PUT 'WARNING: Grouped AE with a wrong outcome. ' _ALL_;
    * Update TEAE to non-TEAE when the following conditions are meet;
       /* previous AE is non-treatment emergent or the treatment has changed */
    IF     ( trtemfl = 'N' OR last_extrt ~= extrt )
       /* intensity/grade not worsened */
       AND last_aesev >= aesev > ' '
       /* not changed to a serious event */
       AND ( last_aeser = aeser OR aeser = 'N' )
       /* AE is not drug-related */
       AND aerel IN('N' 'NOT RELATED' 'UNLIKELY RELATED') THEN DO;
      trtemfl= 'N';
    END;
  END;
  OUTPUT;
  /* save the values to compare it with the next record */
  last_aeterm= aeterm;
  last_aegrpid= aegrpid;
  last_aeout= aeout;
  last_aeser= aeser;
  last_aesev= aesev;
  last_aerel= aerel;
  last_aeout= aeout;
  last_trtemfl= trtemfl;
  last_extrt= last_extrt;
RUN;
PROC SORT DATA=teae08_improvedaes OUT=teae09_sorted;
  BY usubjid aeseq;
RUN;
```

Some studies using the CTCAE grade instead of the AE severity. In such cases, the variable AESEV must be interchanged with the variable AETOXGR. For the given sample data, the first grouped AE (headache) remains treatment emergent because the first part of the AE (AESEQ = 1) starts during treatment “A” and improves during the same treatment. The second grouped AE (Back pain) started as TEAE during treatment “A” but improves during treatment “B”. The second part of the AE is set to non-treatment emergent because the AE improves during another treatment exposure and is therefore not counted as a new AE.

Table 3: Joined data with TEAE flag considering grouped AEs

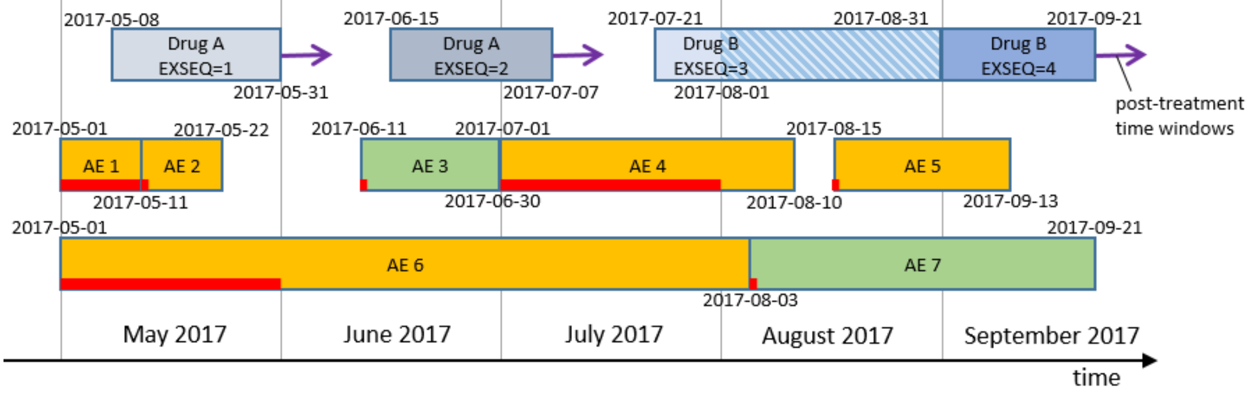


TEAE09_SORTED

AE data								EX data			
Unique Subject Identifier	Sequence Number	Group ID	Reported Term for the Adverse Event	Severity/ Intensity	Start Date/Time of Adverse Event	End Date/Time of Adverse Event	Treatment Emergent Analysis Flag	Sequence Number	Name of Treatment	Start Date/Time of Treatment	End Date/Time of Treatment
USUBJID	AESEQ	AEGRPID	AETERM		AESTDTC	AEENDTC	TRTEMFL	EXSEQ	EXTRT	EXSTDTC	EXENDTC
ABC-1001	1	1	Headache	MODERATE	2017-05	2017-05-11	Y	1	A	2017-05-08T08:20	2017-05
ABC-1001	2	1	Headache	MILD	2017-05-11	2017-05-22	Y	1	A	2017-05-08T08:20	2017-05
ABC-1001	3		Fever	MODERATE	2017-06-11	2017-06	N	.			
ABC-1001	4		Bone pain	MODERATE	2017-07	2017-08-10	Y	2	A	2017-06-15	2017-07-07
ABC-1001	4		Bone pain	MODERATE	2017-07	2017-08-10	Y	3	B	2017-07-21	2017-08
ABC-1001	5		Cold	MILD	2017-08-15	2017-09-13	Y	4	B	2017-08	2017-09-21
ABC-1001	6	2	Back pain	MODERATE	2017-05	2017-08-03	Y	1	A	2017-05-08T08:20	2017-05
ABC-1001	7	2	Back pain	MILD	2017-08-03		N	4	B	2017-08	2017-09-21

The correction is visible in the time diagram in Figure 6. The AE 6 started during treatment “A”, improves with AE 7 during treatment “B”.

Figure 6: Timeline diagram with TEAE flag considering grouped AEs



Step 6: Store all necessary data in ADAE

The last step of the IBA is to make the TEAE flag available for the analysis. To support a review of the data and to carry out further analysis, it is a good practice to store some temporary variables that were necessary to derive the TEAE flag. The detection of treatment emergent is suitable as a first step to create the adverse event analysis dataset ADAE. Therefore, it is assumed that ADAE is not available before this step.

```
/* pre-sort the original AE data */
PROC SORT DATA=sdtm.ae OUT=ae;
  BY usubjid aeseq;
RUN;

/* merge all needed TEAE variables to the AE data */
DATA adae01_teae;
  MERGE ae
        teae09_sorted( IN=a
                       KEEP= usubjid aeseq trtemfl aestdt_min aeendt_max
                             exseq extrt exstdtc exendtc exstdt_min );
  BY usubjid aeseq;
  /* Set AEs to treatment emergent for subjects without any non-missing dates */
  IF NOT a THEN trtemfl= 'Y';
RUN;
```

```
/* Set analysis the start and end date */
DATA adae02_adt;
  SET adae01_teae;
  ATTRIB astdtm FORMAT=is8601dt. LABEL='Analysis Start Datetime';
  ATTRIB astdtf LENGTH=$1        LABEL='Analysis Start Date Imputation Flag';
  ATTRIB asttmf LENGTH=$1        LABEL='Analysis Start Time Imputation Flag';
  ATTRIB aendtm FORMAT=is8601dt. LABEL='Analysis End Datetime';
  ATTRIB aendtf LENGTH=$1        LABEL='Analysis End Date Imputation Flag';
  ATTRIB aentmf LENGTH=$1        LABEL='Analysis End Time Imputation Flag';
  /* Set analysis start date as the maximum of aestdt_min and exstdt_min */
  astdtm= max(aestdt_min,exstdt_min);
  astdtf_temp= compare(put(astdtm,is8601dt.),aestdtc);
  IF astdtf_temp<5 THEN DO;
    astdtf= 'Y';
    asttmf= 'H';
  END;
  ELSE IF astdtf_temp<8 THEN DO;
    astdtf= 'M';
    asttmf= 'H';
  END;
  ELSE IF astdtf_temp<11 THEN DO;
    astdtf= 'D';
    asttmf= 'H';
  END;
  ELSE IF astdtf_temp<14 THEN DO;
    astdtf= '';
    asttmf= 'H';
  END;
  ELSE IF astdtf_temp<17 THEN DO;
    astdtf= '';
    asttmf= 'M';
  END;
  ELSE IF astdtf_temp<20 THEN DO;
    astdtf= '';
    asttmf= 'S';
  END;
  /* <<< repeat the code to set the analysis end date as aeendt_max >>> */

  DROP aestdt_min aeendt_max exstdt_min astdtf_temp aendtf_temp;
RUN;
```

The dataset ADAE02_ADT is a good starting point for further derivations of ADAE. With this IBA the TEAE flag and the analysis start date can be computed with partial and complete missing dates, as well as overlapping treatment exposure intervals. In addition, longer interruptions are also considered. Moreover, the analysis of adverse events can become very complicated. The next section will describe some additional thoughts that should be considered.

THINGS TO CONSIDER

After the IBA an AE can lead to multiple records in case the AE persist over different study treatments. To prevent errors by the analysis of overall counts the first occurrence of an AE should be flagged with the OCCDS occurrence flag variables as described in the document CDISC ADaM Occurrence Data Structure (OCCDS) Version 1.0, section 3.2.6.

Another topic is the analysis of the AE duration. For the duration calculation, you need to consider more special scenarios. A good instruction for AE duration calculation is described in "*Considerations in ADaM Occurrence Data: Handling Crossover Records for Non- Typical Analysis*" published by Karl Miller and Richann Watson.

In some studies, it is possible to detect the chronological order of the exposure records beside partial and complete missing dates. In those special cases, an additional step can be added after step 2. Each start date/time must be equal or after than all previous start date/times (*STDT_MIN) of the same treatment. Conversely, each end date/time must be equal or earlier as all following end date/times (*ENDT_MAX) of the same treatment. It is advisable to carry out a check if the data is arranged in a chronological order.

Another special case is when an AE started pre-treatment but ends during treatment with fatal. In those cases, the study team should check if the death is related to the study drug and if an additional observation to present a worsening is needed.

CONCLUSION

This paper shows a straightforward algorithm to create the TEAE flag which considers all special cases shown in "*TEAE, Did I flag it right?*" by Arun Raj Vidhyadharan and Sunil Mohan Jairath and "*How to define Treatment Emergent Adverse Event (TEAE) in crossover clinical trials?*" by Mengya Gillian Yin and Wen Tan. The algorithm also provides the ability to deal with longer study treatment interruptions. The TEAE flag is the best starting point for the analysis dataset ADAE. For duration calculations of an AE, it can be needed to extend the ADAE dataset, especially for crossover trials. It is also a good practice to discuss all special cases with the study team on a regular basis. That includes AEs which persists while the treatment is changed, complete missing dates, as well as data issues related to AE and exposure data. Beside that with this paper the TEAE detection runs more stable than ever before.

REFERENCES

Arun Raj Vidhyadharan and Sunil Mohan Jairath, "*TEAE, Did I flag it right?*", PharmaSUG 2015 - Paper PO01, https://www.pharmasug.org/proceedings/2015/PO/PharmaSUG-2015-PO01.pdf

CDISC Analysis Data Model Team, "*CDISC ADaM Implementation Guide*" Version 1.1 from 12-Feb-2016, CDISC, https://www.cdisc.org/standards/foundational/adam

CDISC Analysis Data Model Team, "*CDISC ADaM Occurrence Data Structure (OCCDS)*" Version 1.0 from 12-Feb-2016, https://www.cdisc.org/standards/foundational/adam

CDISC Submission Data Standards Team, "*CDISC SDTM Implementation Guide*" Version 3.2 from 26-Nov-2013, CDISC, https://www.cdisc.org/standards/foundational/sdtmig

ICH Expert Working Group, "*ICH HARMONISED TRIPARTITE GUIDELINE - STATISTICAL PRINCIPLES FOR CLINICAL TRIALS E9*" Current Step 4 version from 05-Feb-1998, ICH, http://www.ich.org/fileadmin/Public_Web_Site/ICH_Products/Guidelines/Efficacy/E9/Step4/E9_Guideline.pdf

Karl Miller and Richann Watson, "*Considerations in ADaM Occurrence Data: Handling Crossover Records for Non-Typical Analysis*", PharmaSUG 2015 - Paper DS06, https://www.pharmasug.org/proceedings/2015/DS/PharmaSUG-2015-DS06.pdf

Mengya Gillian Yin and Wen Tan, "*How to define Treatment Emergent Adverse Event (TEAE) in crossover clinical trials?*", PharmaSUG 2017 - Paper IB03, https://www.pharmasug.org/proceedings/2017/IB/PharmaSUG-2017-IB03.pdf

ACKNOWLEDGMENTS

Many thanks to Juergen Lembke, Murat Ipek and the PhUSE data handling stream chairs John Moone and Joerg Guettner for their review of this paper.

CONTACT INFORMATION

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

Matthias Lehrkamp
Bayer AG
Email: matthias.lehrkamp@bayer.com