Paper 206-2023

Generation of AE (Adverse Events) summary tables by worst CTC Grade utilizing SAS®

Ballari Sen, Bristol Myers Squibb

ABSTRACT

Adverse event (AE) analysis is an essential piece in the safety assessment, and AEs are collected in almost every trial and clinical study report [1]. Hence, it is essential that we are clear and very careful while creating these tables and displaying correct counts.

AE summary tables by worst CTCAE (Common Terminology Criteria for Adverse Events) grades are tabulated and programmed using MedDRA System Organ Class (SOC) and Preferred Term (PT), and each subject is counted only once within SOC and within PT. Sorting is by descending frequency order of SOC and PT within a treatment arm of interest.[1]. AE tables with additional CTCAE sub-categories (e.g., Grades 3-4, Grade 5), percentage cuts (e.g., >= 5% frequency), and long MedDRA dictionary text (i.e., up to 40 characters) creates additional programming complexity [2].

This paper presents a practical approach for generating different types of AE toxicity grade tables using the SAS® software approach. A set of guidelines is presented to simplify the programming process. The methodology aims to facilitate the analysis and interpretation of AE data, enabling researchers and clinicians to make informed decisions regarding patient safety. This paper will further discuss this approach in detail and share the code in getting the work done in an efficient way. This paper also addresses some of the programming validation edit-checks which will further help to cross-check the counts that are generated and displayed.

INTRODUCTION

DESCRIPTION

AE summary or count tables are very essential for physicians and pharmaceutical companies to assess the safety profile of the study drug. These tables and outputs usually output the number of adverse events, the number of patients in each treatment group in whom the event occurred. Typically, adverse events are grouped by System Organ Class, Preferred Terms and/or other variables of interest [3].

They further help to summarize organ toxicity data in a clear and concise manner. The grade details summarized in tables and outputs utilizes the CTCAE (Common Terminology Criteria for Adverse Events) for organ toxicity. The CTCAE defines as any abnormal clinical finding temporally associated with the use of a therapy as an adverse event and standardizes adverse events in five grades by the National Institutes of Health. Toxicity is graded as mild (Grade 1), moderate (Grade 2), severe (Grade 3), or life-threatening (Grade 4) and Death (Grade 5) [2].

Depending on the study design, adverse event summary by grade tables can display all five toxicity grades or only selected grades, such as only grade 2 or grade 3. Therefore, it is vital for us to know what type of AE table is required, and to adjust the program according to the output being generated [2].

Prior to generating the output tables, it is very important that we are aware of the adverse event analysis, adverse event attribute and coding of adverse events that will be very useful and informative while creating these tables [3]. The paper will address this in the first section and will also generate SAS programs to accommodate and produce accurate tables to different types of AE grades which will help explain the clinical data and summarize it right, therefore in all a good programming approach is conducted to provide a high-quality AE table for final Clinical study report.

PROCEDURES

The following guidelines are performed to accomplish the AE Summary Tables.

ADVERSE EVENT ANALYSIS

The safety evaluation of a clinical trial includes the analysis of adverse events.

An adverse event is defined as:

Any untoward medical occurrence in a patient or clinical investigation subject administered a pharmaceutical product and which does not necessarily have to have a causal relationship with this treatment.

This definition of an adverse event (AE) includes any unfavorable and unintended sign, symptom, or disease that is temporally associated with the use of an investigational product, regardless of whether the AE is considered to be related to the product [3].

CODING OF ADVERSE EVENTS

Medical Dictionary for Regulatory Activities (MedDRA) is a widely used global standard for coding of adverse events.

It is also recommended that all levels of terms in the MedDRA: System Organ Class (SOC), High Level Group Term (HLGT), High Level Term (HLT), Lowest Level Term (LLT), and Preferred Term (PT) are represented, as these are frequently useful in further analyses of AEs [3].

STEPS BEFORE PROGRAMMING

A well outlined program will save programming time. Several key points should be considered before creating an AE summary by grade table.

The most frequently used method for the comparison of adverse events between treatment groups is the summarization of the number of subjects who experienced a given adverse event at least once by the dictionary derived term (System Organ Class and Preferred Term). These counts and related percentages are given outputs for levels of the System Organ Class and preferred term [3]. The denominator used for the calculation of the percentages is often determined by a population flag such as safety population flag (e.g., SAFFL='Y'), since some subjects exposed to the study drug treatments may not have any adverse events and therefore these subjects would not be presented in the SDTM AE Domain and in the ADaM ADAE analysis dataset. Therefore, the values of the denominator or N Count usually needs to be populated from the ADSL (subject level analysis dataset) dataset rather than ADAE dataset [3].

Program Header

Program headers are created to provide a high-level information about the program. Also, it includes traceability about the program authors and reference to programs.

```
/*-------------------------------------------------------------------------------------------------------------------------------------------/
/ Program:          xx-ae-aecat-xxxx          <Program Root Name>
/ Programmer:      Ballari Sen               <Author BMS User ID>
/ Date:            20221203                  <Original Creation Date use yyyymmdd>
/ Project/Study:   xx/xxx                    <Project or Study>
/ Purpose:    SAS program used to generate tables
                Summary of Serious Adverse Events
                by Worst CTC Grade (Any Grade, Grade 3-4, Grade 5)
/ Modification History:
/ Date     Version Programmer     Description
/ -------- ------- ---------- ----------------------------------
/ 20221202 V01     Ballari Sen    initial version, copied from
```

/ /xxx/prod/xxx/xxx/xx/xx/xxx/xxxx/xxxx/xxx

/- -/

/--/

/ Copyright: Bristol Myers Squibb

/--*/

Program to Obtain events/subject counts

N = xx: This is the safety population count and has to be obtained from ADSL per treatment.

N = No. of subjects in the population.

The below piece of code is used to obtain N count subjects for TRT01a treatment groups.

```
proc sql;
   create table adsl_ as
      select usubjid, trt01a, trt01an
      from xxxx.adsl
      /*---/
      / Subset the population dataset to keep treated subjects
      /---*/
    /*---
   / Populating BIG_N as N count by distinct usubjid
     /---*/
      where trt01a is not missing
      order by usubjid;
   create table Big_N as
     select count (distinct usubjid) as N, trt01a, trt01an as TRTAN
      from adsl_
  group by trt01a, trt01an
  order by trt01an;
quit;
run;
```

The next step is to select some of the ADAE variables from ADAE dataset : USUBJID (Unique Subject Identifier) , TRTA (Actual Treatment) , TRTAN (Actual Treatment (N)) , SAFFL (Safety Population Flag) , AEBODSYS (Body System or Organ Class) , AEDECOD (Dictionary-Derived Term) , ASTDT (Analysis Start Date) , AENDT (Analysis End Date) , AETOXGRN (Standard Toxicity Grade (N)) , APHASEN (Phase (N)) , TEAE100 (Treatment Emergent AE 100 Days Follow-up). The following conditions are applied to select values only where “Treatment Emergent AE 100 Days Follow-up flag” has “Yes” values and Standard Toxicity Grade (N) is not null and safety population flag is “Yes”.

```
/****Creating a dataset from ADAE Dataset by selecting distinct: usubjid,
TRTA, TRTAN, TEAE100, AEBODSYS, AEDECOD, AETOXGRN and populating a where
condition where TEAE100 = "Y" and AETOXGRN ne. and SAFFL = "Y" and TRTAN ne
null. ***********/

proc sort data=xxxx.adae out=adae;
 by usubjid;
run;

/**Merging adae and adsl_ dataset by usubjid*****/

   data merge_adae_adl;
     merge adae  (in=a)
           adsl_ (in=b);
```

```
  by usubjid;
  if a and b;
run;

data adae (keep = usubjid aebodsys aedecod aetoxgrn aphasen
           astdt aendt trta trtan TEAE100 saffl);
  set merge_adae_adl ;
  where TEAE100 = "Y" and aetoxgrn ne . and saffl = "Y"
  and trtan ne.;
run;
```

Further we would need to calculate the “Total no. of subjects with any event count” by System Organ Class and Preferred Term for all the treatment groups and also output their respective toxicity grades. The grade display can be displayed as “Grade = 2” or “Grade = 3-4” or “Grade = 5”. This will produce a table that displays the system organ class , preferred term, treatment groups , toxicity grades and also populate sub-categories for each subject by their toxicity grades. i.e., if AETOXGRN = “3,4” then sub-category is “2”.

```
/*---/
 / Selecting worst tox grade for subject, soc, aedecod;
 Creating a distinct usubjid count for any grade, grade 3-4 & grade
 5 for: total no. of subject with any event
/---*/

    proc sort data=adae_ out=adae_adl_any_grade;
     by usubjid descending aetoxgrn;
    run;

    data adae_adl_any_grade_;
      set adae_adl_any_grade;
      by usubjid descending aetoxgrn ;
      if first.usubjid;
      subcat = 1 ; output ;
      if aetoxgrn in (3, 4) then subcat=2; output;
      if aetoxgrn = 5 then subcat=3; output;
    run;
```

Using the PROC SORT procedures to sort the dataset by descending AETOXGRN within each USUBJID and AEBODSYS and AEDECOD combination. The below code creates two dataset WorstSOC and Worstpt to populate sub-categories for each subject by their AEBODSYS variable and also AEDECOD variable. The code will assign subcat variable as “1” for the first occurrence of each AEBODSYS variable and also AEDECOD variable value and then subcat as “2” where aetoxgrn is 3 or 4 and subcat as “3” if aetoxgrn is 5. The by statement ensures that the dataset is processed in the specified order.

```
/*---/
/ Creating a distinct bodsys_count & aedecod_count for
  any grade, grade 3-4 & grade 5
  /---*/

  proc sort data=adae_ out=merge_adae_adl_bod ;
    by usubjid aebodsys descending aetoxgrn;
  run;

  data WorstSOC;
    set merge_adae_adl_bod;
      by usubjid aebodsys descending aetoxgrn;
       if first.aebodsys ;subcat=1 ; output  WorstSOC ;
```

```
      if aetoxgrn in (3, 4) then subcat=2; output WorstSOC;
      if aetoxgrn = 5 then subcat=3; output WorstSOC;
run;

proc sort data=adae_ out=merge_adae_adl_d;
  by usubjid aebodsys aedecod descending aetoxgrn;
run;

data Worstpt;
  set merge_adae_adl_d;
    by usubjid aebodsys aedecod descending aetoxgrn;
      if first.aedecod ;subcat=1; output Worstpt ;
      if aetoxgrn in (3, 4) then subcat=2; output Worstpt;
      if aetoxgrn = 5 then subcat=3; output Worstpt;
run;
```

The code below calculates the distinct subject counts for the number of subjects experiencing any adverse event (system organ class and preferred term) for the mentioned AETOXGRN grade numbers in ("Any Grade"," Grade 3-4"," Grade 5") and for their respective treatment groups within the study.

```
/*---/
/ Populating distinct usubjid counts for total no.
  of subject with any event for all the grades
  and ordering them as ord_1
  /---*/

  proc sql;
    create table all_C_total_sub_event as select
      distinct 1 as ord_1, subcat,trtan,count (unique usubjid) as count
    from adae_adl_any_grade_
      group by subcat,trtan;
  quit;

  proc sort data=all_C_total_sub_event;
    by trtan;
  run;
```

A merge is created between "N" Count for each category and the "total no. of subjects with any event" count to populate the percentage for the "total no. subject count for any event" with their grade number and the treatment groups within the study. The percentage is calculated by dividing the event_count by the corresponding "N count".

```
/*---/
 / Creating a merge between Big_N count and
   All_C_Total_Sub_event dataset merge to populate the percent
   as count * 100 /N, populating category
   as "Total Subjects with an Event"
/---*/

    data percent_;
      merge Big_N all_C_total_sub_event;
      length category $100. per_ $30.;
      by TRTAN ;
      percent=put(count * 100/N,5.1);
      per_=(put(count,3.))||' ('||percent||') ';
      category="TOTAL SUBJECTS WITH AN EVENT";
      drop percent;
```

```
        run;
```

Body system count and preferred term count are calculated and are the number of events for that particular SOC and PT. They are further ordered as 1 and 2. These counts are the number of events a subject has encountered after taking treatment drug and we do not count it as unique event. This step utilizes the SQL to calculate the count of unique subjects for each combination of aebodsys , aebodsys subcat variables and treatment groups.

```
 /*---/
   / System Organ Class Population
    / Preferred Term Population
/---*/

/*---/
 / Creating a distinct usubjid bodsys count
   for all the toxicity grades and ordering
   them as ord_1: 1 and ord_2: 2.
/---*/

   proc sql;
     create table bdysys_count_all as select 1 as ord_2 ,
     2 as ord_1,aebodsys,subcat,trtan,count(unique usubjid) as count
     from WorstSOC group by aebodsys, subcat, trtan;
   run;

/*---/
  / Creating a distinct usubjid aedecod count
    for all the grades and ordering them 2 as ord_1 and 2 as ord_2
/---*/

  proc sql;
    create table decod_count_all as select 2 as ord_2 ,
    2 as ord_1,aebodsys,aedecod,subcat,trtan,count(unique usubjid) as count
    from worstpt group by aebodsys, aedecod, subcat, trtan;
  run;

/*---/
  / Sorting bdysys_count_all and decod_count_all
    by trtan before merging with Big_N for percentage generation
/---*/

    proc sort data=bdysys_count_all ;
     by trtan ;
     run;

     proc sort data=decod_count_all;
       by trtan ;
     run;
```

The frequency cut-off with extended follow-up which was 5 % was programmed by populating records for subjects with the preferred term and the body system count for all the grades and the treatment groups by creating the percentage by N counts for all the subjects with treatment groups.
Finally selecting only those SOC and PT values where the percentage is greater than 0.05 value.

```
 /*---/
  / 5% preferred term cutoff for aedecod and for aebodsys variables
 /---*/
```

```
    data percent_decod_cutoff;
        merge Big_N decod_count_all ;
        length per_ $30.;
        by trtan ;
        percent=(count*100 /N);
        p_round=put(percent, 5.2);
        per_=compress(put(count,10.))
             || ""  || "("  ||compress(put(percent,5.1)) || ")";
        drop percent ;
         if subcat eq 1 then
          pct=count/N;
         if subcat eq 2 then
          pct=count/N;
         if subcat eq 3 then
          pct=count/N;
         if pct >=0.05;
        drop pct ;
       run;

/*---/
  / Selecting only distinct body system organ class and preferred term
    Variable values following the 5% cut off.
 /---*/
       proc sort data=percent_decod_cutoff
         out=cut_soc (keep=aebodsys)nodupkey;
         by aebodsys;
       run;

       proc sort data=percent_decod_cutoff
         out=cut_pt (keep=aebodsys aedecod) nodupkey;
         by aebodsys aedecod;
       run;
```

To populate only the AEBODSYS and AEDECOD variables and counts for subjects following the 5% cut off, a merge is programmed between the bodsyscount, preferred term counts for subjects for their particular SOC and PT and the dataset where only the distinct AEBODSYS & AEDECOD variable are created following 5 % cut off. The merged dataset now has only records of subjects having number of events for any particular SOC and PT and are following the extended 5 % cut off according to the DPP specifications.

```
/*---/
  / To include all the aebodsys count with 5 % Cutoff record
/---*/
      proc sort data=bdysys_count_all ;
       by aebodsys ;
      run;

      data aebodsys_include;
        merge bdysys_count_all (in=in_soc)
        cut_soc(in=cut_soc_) ;
        by aebodsys ;
        if in_soc and cut_soc_;
      run;

/*---/
    / To include all the aedecod count with 5 % Cutoff record
      /---*/
```

```
proc sort data=decod_count_all ;
 by aebodsys aedecod ;
run;

data aedecod_include;
  merge decod_count_all(in=in_aedecod)
        cut_pt (in=cut_put_);
  by aebodsys aedecod ;
  if in_aedecod and cut_put_;
run;
```

A merge is created between “N” count for each treatment category and aebodsys and aedecod counts for grouped by treatment category and subcategorized by toxicity grades (with 5% cut-off), they are further sorted, transposed(from vertical to horizontal data structure) and merged by their sub-categories created earlier to represent the mock-shell for the adverse event summary table output. The percentage is calculated by dividing the event_count by the corresponding “N count”.

```
/*---/
   / Creating a merge between Big_N count
     and aebodsys_include & aedecod_include for populating percentages
 /---*/

    proc sort data=aebodsys_include ;
      by trtan ;
    run;

    data percent_bodsys;
      merge Big_N aebodsys_include ;
      length per_ $30.;
      by trtan ;
      percent=count * 100 /N;
      per_=(put(count,3.))||' ('||(put(percent,5.1))||') ';
      category = AEBODSYS ;
      drop percent ;
    run;

    proc sort data=aedecod_include ;
      by trtan ;
    run;

    data percent_decod ;
      merge Big_N aedecod_include ;
      length per_ $30.;
      by trtan ;
            percent=count * 100 /N;
            per_=(put(count,3.))||' ('||(put(percent,5.1))||') ';
            category = AEDECOD ;
            drop percent ;
          run;
```

A merge is created between “total number of subjects with any event count and percentage”, “subjects for number of events for any SOC and PT counts and percentages”, they are further sorted, transposed(from vertical to horizontal data structure) and merged by their sub-categories created earlier to represent the mock-shell for the adverse event summary table output.

```
/*---/
    / Creating a merge between percent_ percent_bodsys percent_decod
      to join all the percentage from total no. subjects
      with an event, aebodsys_percent & aedecod_percent
 /---*/

   data all_event_bodsys_decod ;
     set percent_ percent_bodsys percent_decod_1 ;
   run ;

/*---/
    / Sort the datasets before transposing the datasets
  /---*/

proc sort data = all_event_bodsys_decod
          out = all_event_bodsys_decod_ ;
  by ord_1 ord_2 category aebodsys aedecod
     subcat trtan ;
run ;

/*---/
    / Creating a transpose for each treatment groups,
      toxicity grade categories into trt1,trt2 & trt3/
/---*/

     proc transpose data= all_event_bodsys_decod_
   out=all_event_bodsys_decod_1 prefix=trt;
      by ord_1 ord_2 category aebodsys
         aedecod subcat;
      var per_ ;
      id trtan ;
   run;

   proc transpose data=all_event_bodsys_decod_1
   out=all_event_bodsys_decod_1_1 prefix=trt1_;
     by ord_1 ord_2 category aebodsys aedecod;
     var trt1;
     id subcat ;
   run;

   proc transpose data=all_event_bodsys_decod_1
   out=all_event_bodsys_decod_1_2 prefix=trt2_;
     by ord_1 ord_2 category aebodsys aedecod;
     var trt2;
     id subcat ;
         run;
```

Further renaming the variables with respect to the treatment groups and toxicity grades.

```
/*---/
    /  Merging  : all_event_bodsys_decod_1_1 ,
       all_event_bodsys_decod_1_2 & all_event_bodsys_decod_1_3
       and creating the required the treatment variables.
  /---*/

data merge_all ;
  merge all_event_bodsys_decod_1_1
```

```
  all_event_bodsys_decod_1_2 all_event_bodsys_decod_1_3 ;
  by ord_1 ord_2 category aebodsys aedecod;
run ;

data merge_all_f (rename=(trt1_1=XXXXXXX_Any_Grade trt1_2=XXXXX_3_4
trt1_3=XXXX_5 trt2_1=XXXXX_XXXXX_Any_Grade
trt2_2=XXXXX_XXXXX_3_4 trt2_3=XXXXX_XXXXX_5
trt3_1=XXXXXX_Any_Grade trt3_2=XXXXXXX _3_4
trt3_3=XXXXXXX_5));
  set merge_all ;
  if trt1_1 = '' then trt1_2 = '  0' ;
  ……    ;
  ……   ;
run;
```

Display the layout of the adverse event table summary

The order of the category for outputs are critical. In the overall AE output summary by worst CTC grades counts will be displayed by the order from categories defined in the shell. It is very easy and programmable to set up the category order by marking the sorting keys and order on paper before programming.

For the AE summary table by system organ class and preferred term, the sorting key could be in either alphabetical order or in frequency order by columns. The programmer needs to know which column is headmost for sorting. For e.g., Total Subjects with an event would be the first, followed by System Organ class and Preferred term variables following descending sorting order sequence of subject variable counts for SOC and PT.

The final output table is programmed utilizing PROC REPORT procedures and through internal macro for generating zero-page number in the output. The header and appropriate footnotes should be populated after verifying with DPP specifications.

```
/*---/
    / Sorting and ordering by xxxxx_Any_Grade_N
      xxxxxx_3_4_N xxxxx_5_N aebodsys and aedecod
      to order the table output
  /---*/

    data merge_all_f ;
      set merge_all_f ;
      xxxxx_Any_Grade_N =input(scan(compress(xxxxx_Any_Grade,'
    ','kdp'),1),best.) ;
      xxxxxxx_3_4_N =input(scan(compress(xxxxx_3_4,' ','kdp'),1),best.) ;
      xxxxxx_5_N =input(scan(compress(xxxxxx_5,' ','kdp'),1),best.) ;
    run ;

    proc sort data = merge_all_f out = merge_all_f_ ;
     by ord_1 DESCENDING xxxxx_Any_Grade_N  ;
    run ;

    data merge_all_f_1 (keep = aebodsys ord_1 ord_2 xxxxx_Any_Grade_N
    xxxxx_3_4_N xxxxx_5_N) ;
      set merge_all_f_ ;
      where ord_2 = 1 and ord_1 = 2 ;
    run ;

    data merge_all_f_2 ;
     set merge_all_f_1 ;
```

```
     n+1;
    run ;

    proc sql;
      create table merge_all_f_3 as select a.*,b.N
      from merge_all_f_ a left join merge_all_f_2 b
      on a.aebodsys = b.aebodsys ;
    quit ;

    data merge_all_f_3_ ;
      set merge_all_f_3 ;
      if category = "TOTAL SUBJECTS WITH AN EVENT" then n = 0;
    run ;

    proc sort data = merge_all_f_3_ out = merge_all_f_3_1 ;
     by n ord_2 descending xxxxx_Any_Grade_N
     descending xxxxxx_3_4_N descending xxxxx_5_N ;
    run ;

/*---/
  / Generating the Proc Report and the %pagenumz macro
    /---*/

    options center nonumber nodate ls=132 ps=45 missing=' '
    nobyline mprint mlogic symbolgen;
    title1 j=c "Summary of Serious Adverse Events by Worst CTC Grade";
    title2 j=c "(Any Grade, Grade 3-4, Grade 5) with Extended Follow-up
    with 5% Cutoff";
    title3 j=c "All Treated Subjects";
    %LET N1 = ....;
    %LET N2 = ....;
    %LET N3 = ....;

    proc report data=merge_all_f_3_1 nowd missing split="|" headline
    headskip center formchar="|-----------|+|---+=|-/\<>" style=monospace
    ls=132 ps=45 spacing=0 ;
    column N ord_2 category("xxxx|N = &N1|--" xxxx_Any_Grade xxxx_3_4
    xxxx_5) ("xxxx + xx|N = &N2|--" xxxx_xx_Any_Grade xxxx_xx_3_4
    xxxx_xx_5) ("xx|N = &N3|--" xx_Any_Grade xx_3_4 xx_5);

    define N /order order=data noprint;
      define ord_2 /order order=data noprint;
      define category /display "System Organ Class (%)|Preferred Term (%)"
    flow width=24 left;
      define xxxx_Any_Grade /display "Any Grade" flow width=11 spacing=1
    left;
    ……
    ……
      endcomp;

      break after N/skip;

      compute after _page_/left;
          line @1 132*'-';
          line 'MedDRA Version: ----';
          line 'CTC Version ----';
```

```
            line 'Includes events reported between first dose and 100 days
      after last dose of study therapy.';
        endcomp;
      run ;
      title;


/*-------------------------------------------------------------/
/ Macros for adding page numbers and page '0'.
/-------------------------------------------------------------*/

  %pagenumz (ftoutdd="/…./…./…..../…..../…….lst");

  %mpagezero(studyid     = ……,
             doc_id      = ……,
             tableno     = %str(Table ……),
             destination = /…/…./…./….,
             title1      = ……. ;
             title2      = %nrstr((……),
             title3      = ……..,
             status      = FINAL
             orientation = LANDSCAPE
             );

  run ;
```

RESULTS

```
DOCUMENT ID:       xx-ae-aecat-xxx.lst
TABLE NO.:         Table x.x.xx
TITLE:             Summary of Serious Adverse Events by Worst CTC Grade
                   (Any Grade, Grade 3-4, Grade 5) with Extended Follow-up with 5% Cutoff
                   All Treated Subjects
ORIENTATION:       LANDSCAPE
STATUS:            FINAL            orientation = LANDSCAPE
AUTHOR:            Ballari Sen
CREATED:
PROGRAM:
FORMAT:
LOG:
OUTPUT:
SAS VERSION:
OS VERSION:
#END REPORT EXECUTION INFORMATION#
Protocol:                                                                                                    Page 1 of 4
                                        Summary of Serious Adverse Events by Worst CTC Grade
                                  (Any Grade, Grade 3-4, Grade 5) with Extended Follow-up with 5% Cutoff
                                                  All Treated Subjects

-------------------------------------------------------------------------------------------------------------------------------
                                      xxxxxxxxx                    xxxxxxxxx + xxxxxxxxx                      xxxxxxxxx
                                       N = xxx                             N = xxx                            N = xxx
System Organ Class (%)   ---------------------------------- ---------------------------------- ----------------------------------
Preferred Term (%)       Any Grade   Grade 3-4   Grade 5     Any Grade   Grade 3-4   Grade 5     Any Grade   Grade 3-4   Grade 5
-------------------------------------------------------------------------------------------------------------------------------

TOTAL SUBJECTS WITH AN   xxx ( xx.x) xxx ( xx.x)  xx ( xx.x) xxx (xxx.x) xxx ( xx.x)  xx ( xx.x) xxx ( xx.x) xxx ( xx.x)  xx ( xx.x)
EVENT

GENERAL DISORDERS AND    xxx ( xx.x)  xx (  x.x)   x (  x.x) xxx ( xx.x)  xx ( xx.x)   x (  x.x) xxx ( xx.x)  xx (  x.x)   x (  x.x)
ADMINISTRATION SITE
CONDITIONS
  FATIGUE                xxx ( xx.x)   x (  x.x)   x         xxx ( xx.x)  xx (  x.x)   x         xxx ( xx.x)   x (  x.x)   x
  PYREXIA                 xx ( xx.x)   x (  x.x)   x         xxx ( xx.x)   x (  x.x)   x          xx ( xx.x)   x (  x.x)   x
-------------------------------------------------------------------------------------------------------------------------------
MedDRA Version: xx.x
CTC Version x.x
Includes events reported between first dose and 100 days after last dose of study therapy.
```

Figure 1. Summary of Serious Adverse Events by Worst CTC Grade.

CONCLUSION

Maintaining high efficiency without compromising quality is a necessity in the pharmaceutical industry. This paper gives a detailed explanation about ADAE and ADSL and also describes an efficient programming method for generating AE Summary table by system organ class and preferred term. AEs play a crucial role in assessing the safety of interventions in clinical trials and analyzing them by CTC grade provides valuable insights into their severity. This approach contributes to assessing the safety profile of a particular intervention or drug treatment and further summarizing AE data by worst CTC grade , leveraging the capabilities of SAS.

REFERENCES

1. Hu, Jiannan. 2011.“A Practical and Efficient Approach in Generating AE (Adverse Events) Tables within a Clinical Study Environment”. *NESUG 2011*.
2. Yang, Zhengxin (Cindy). 2015.“A Practical Approach to Create Adverse Event Summary by Toxicity Grade Table”. *PharmaSUG 2015 - Paper QT45*.
3. Landge, Rucha. 2016. “AE: An Essential Part of Safety Summary Table Creation.”. *PharmaSUG 2016 - Paper IB11.*

ACKNOWLEDGMENTS

I would like to thank my managers, Ankur Sharma and Todd Rider for their constant encouragement and valuable and constructive feedback. I would also like to thank Samar Noor for providing me the opportunity to present the paper at WUSS 2023 Conference.

CONTACT INFORMATION

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

Name : Ballari Sen
Company : Bristol Myers Squibb
Email : Ballari.sen@bms.com