```
*******************************************************************************************;
**  Program Name    :  adae-s130-sae-unb2-ped6.sas                              **;
**  Date Created    :  15Nov2021                                       **;
**  Programmer Name :  (b) (4), (b) (6)                                       **;
**  Purpose         :  Create adae-s130-sae-unb2-ped6                             **;
**  Input data      :  adae adsl                                 **;
**  Output data     :  adae-s130-sae-unb2-ped6.html                               **;
*******************************************************************************************;
options mprint mlogic symbolgen mprint symbolgen mlogic nocenter missing=" ";
**Setup the environment**;
%let
bprot=/Volumes/app/cdars/prod/sites/cdars4/prjC459/nda2_unblinded_esub/sbla1215_esub_adam/saseng/cdisc3_0/;
%let prot=/Volumes/app/cdars/prod/sites/cdars4/prjC459/nda2_unblinded_esub/sbla1215_esub_adam/saseng/cdisc3_0;
libname datvprot "&bprot.data_vai" access=readonly;
%let codename=adae-s130-sae-unb2-ped6;

%let outlog=&prot./analysis/eSUB/logs/&codename..log;
%let outtable=&prot./analysis/eSUB/output/&codename..html;

proc printto log="&outlog." new;
run;
****************************************************************************;
*SPECIFICATION 1 -1) gen_cdisc_suminit *;
* 2) null table when analysis dataset is empty *;
* 3) Set Default value to ToT parameters *;
****************************************************************************;

data g_adsl_dsin;
  set datvprot.adsl;

  if trt01an=8 and agegr4n=1 then
    trtarn=1;
  else if trt01an=8 and agegr4n=2 then
    trtarn=2;
  else if trt01an=9 and agegr4n=1 then
    trtarn=3;
  else if trt01an=9 and agegr4n=2 then
    trtarn=4;
  trtar=trt01a;

  if trt01an eq 7 and agegr1n eq 3 then
    trtarn2=1;
  else if trt01an eq 9 and agegr1n eq 3 then
    trtarn2=2;
  else if trt01an eq 7 and agegr1n eq 4 then
    trtarn2=3;
  else if trt01an eq 9 and agegr1n eq 4 then
    trtarn2=4;
  where saffl="Y" and agegr4n eq 1 and phasen in (2, 3, 4)
    and HIVFL ne "Y" and (VAX101DT ne . and BDCSRDT ne .) and (EOSDCDT gt
    input("13MAR2021", date9.) or EOSDCDT eq .);
run;
```

```
data g_a_dsin;
  set datvprot.adae;

  if trt01an=8 and agegr4n=1 then
    trtarn=1;
  else if trt01an=8 and agegr4n=2 then
    trtarn=2;
  else if trt01an=9 and agegr4n=1 then
    trtarn=3;
  else if trt01an=9 and agegr4n=2 then
    trtarn=4;
  trtar=trt01a;

  if trt01an eq 7 and agegr1n eq 3 then
    trtarn2=1;
  else if trt01an eq 9 and agegr1n eq 3 then
    trtarn2=2;
  else if trt01an eq 7 and agegr1n eq 4 then
    trtarn2=3;
  else if trt01an eq 9 and agegr1n eq 4 then
    trtarn2=4;
  analysis_subset='Y';
  where DATCHGFL="Y" and AECAT='ADVERSE EVENT' and saffl="Y" and AESER='Y' and
    agegr4n eq 1 and PHASEN in (2, 3, 4) and vphasen>0
    and .<VAX101DT<=ASTDT<=BDCSRDT and HIVFL ne "Y";
run;

data g_adsl_dsin;
  set g_adsl_dsin;

  if TRT01AN in (8) then
    do;
      newtrtn=1;
      newtrt=coalescec("BNT162b2 (30 (*ESC*){unicode 03BC}g)", TRT01A);
      output;
    end;

  if TRT01AN in (9) then
    do;
      newtrtn=2;
      newtrt=coalescec("Placebo", TRT01A);
      output;
    end;
run;

data g_a_dsin;
  set g_a_dsin;

  if TRT01AN in (8) then
    do;
      newtrtn=1;
      newtrt=coalescec("BNT162b2 (30 (*ESC*){unicode 03BC}g)", TRT01A);
      output;
    end;
  if TRT01AN in (9) then
    do;
      newtrtn=2;
      newtrt=coalescec("Placebo", TRT01A);
      output;
    end;
run;

**************************************************************************************;
*SPECIFICATION 2 -1) determine the no of tiers from g_a_tiervar *;
* 2) any uncoded terms set to uncoded||term label *;
**************************************************************************************;
;

data g_a_dsin1;
  set g_a_dsin;
  where ^missing(aeterm);

  if missing(aedecod) then
    aedecod=aeterm;

  if missing(AEBODSYS) then
    AEBODSYS='**Uncoded Terms Being Queried';

  if missing(AEDECOD) then
    AEDECOD='**Uncoded Terms Being Queried';
  aebodsys=upcase(substr(aebodsys, 1, 1))||strip(upcase(substr(aebodsys, 2,
    (length(aebodsys)-1))));
run;

**************************************************************************************;
*SPECIFICATION 3 -1) Remove Duplicate observations from AE data. *;
* 2) create flags _anyae _event to identify summary counts *;
* for summary rows *;
* 3) merge adsl and analysis dataset *;
**************************************************************************************;

proc sort data=g_a_dsin1 out=_a_dsin nodupkey;
  by newtrtn usubjid aedecod astdt ATOXGR;
run;

proc sort data=_a_dsin;
  by newtrtn usubjid AEBODSYS AEDECOD;
run;

data _a_dsin;
  set _a_dsin;
  by newtrtn usubjid AEBODSYS AEDECOD;

  if first.usubjid then
    _anyae=1;

  if aeterm ne '' then
```

```
    _event=1;
  _uniqid=_n_;
run;

proc sort data=g_adsl_dsin out=_ds1;
  by usubjid newtrtn;
run;

proc sort data=_a_dsin out=_ds2;
  by usubjid newtrtn;
run;

data final;
  merge _ds1(in=d1) _ds2(in=d2);
  by usubjid newtrtn;

  if d1;
run;

data final;
  set final;

  if nmiss(_anyae) then
    _anyae=0;

  if nmiss(_event) then
    _event=0;
run;

******************************************************************************;
*SPECIFICATION 4 -1) minitial - initialize isam *;
* 2) mentry - setup bigN, treatment, by variables & subgroups *;
* 3) apply PCTCUTOFF when G_A_PCTOFF not null *;
******************************************************************************;

data _data1;
  set final;
  where (NEWTRTN is not missing);
run;

proc sql noprint;
  select count(unique NEWTRTN) into :_trtn from _data1 where NEWTRTN is not
    missing;
quit;

proc sort data=_data1;
  by NEWTRTN USUBJID;
run;

data _data1;
  retain _trt 0;
  length _str $200;
  _datasrt=1;
  set _data1 end=eof;
  by NEWTRTN USUBJID;
  drop _str;
  _str=' ';
  _lastby=1;
  _dummyby=0;

  if first.NEWTRTN then
    do;

      if not missing(NEWTRTN) then
        do;
          _trt=_trt + 1;
        end;
      _str=NEWTRT;

      if _trt > 0 then
        call symput('_trtlb'||compress(put(_trt, 4.)), trim(left(_str)));
    end;
run;

*---------------------------------------------------------------------;
* Count number of patients in each treatment. ;
*---------------------------------------------------------------------;

proc sql noprint;
  select compress(put(count(*), 5.) ) into :_trt1 - :_trt2 from (select distinct
    USUBJID, _trt from _data1 where NEWTRTN is not missing) group by _trt;
  select compress(put(count(*), 5.) ) into :_trt3 from (select distinct USUBJID
    from _data1 where NEWTRTN is not missing);
quit;

*---------------------------------------------------------------------;
* Generate a dataset containing all by-variables ;
*---------------------------------------------------------------------;

proc sort data=_data1 out=_bydat1(keep=_datasrt _dummyby) nodupkey;
  by _datasrt;
run;

proc sort data=_data1 out=_data1;
  by _datasrt;
run;

******************************************************************************;
*SPECIFICATION 5 -1) Any AE counts and percent *;
******************************************************************************;

data _null_;
  set _data1 end=eof;

  if eof then
    call symput('dptlab', vlabel(_ANYAE));
run;
```

```
data _anal1;
  length _ANYAE 8;
  set _data1;
  where same and _ANYAE is not missing;
  _blcksrt=1;
  _cnt=1;
  _cat=1;

  if _trt <=0 then
    delete;
  output;
run;

proc sort data=_anal1;
  by _datasrt _blcksrt _ANYAE _trt _cat;
run;

*--- Counts for each by-sequence, dependant var, and treatment combination ---*;

data _temp1;
  set _anal1;
  output;
run;

proc sort data=_temp1 out=_temp91 nodupkey;
  by _datasrt _blcksrt _cat _ANYAE _trt USUBJID;
  ;
run;

proc freq data=_temp91;
  format _ANYAE;
  tables _datasrt*_blcksrt*_cat * _ANYAE * _trt / sparse norow nocol nopercent
    out=_pct1(drop=percent);
run;

proc sort data=_anal1 out=_denom1(keep=_datasrt _cat) nodupkey;
  ;
  by _datasrt _cat;
run;

data _denom1;
  set _denom1;
  by _datasrt _cat;
  label count='count';
  _trt=1;
  count=&_TRT1.;
  output;
  _trt=2;
  count=&_TRT2.;
  output;
run;

*--------------------------------------------------------------------;
* Create _DENOMF a frame dataset for the denominators ;
```

```
*-------------------------------------------------------------------;

data _denomf1;
   _datasrt=1;
   set _bydat1(keep=);
   * All treatment groups ;
   _trt1=0;
   _trt2=0;
   * _CAT is the subgroup variable ;
   _cat=1;
   output;
run;

*-------------------------------------------------------------------;
* Transpose _DENOM into _DENOMIN to get COUNT as _TRTn columns ;
*-------------------------------------------------------------------;

proc transpose data=_denom1 out=_denomin1(drop=_name_ _label_) prefix=_trt;
   by _datasrt _cat;
   var count;
   id _trt;
run;

*-------------------------------------------------------------------;
* Create _FRAME dataset using all combinations of category variable ;
*-------------------------------------------------------------------;

data _frame1;
   _datasrt=1;
   set _bydat1(keep=);
   _blcksrt=1;
   length _ANYAE 8;
   _catLabl=" ";
   _trt=1;
   _ANYAE=1;
   _catord=1;
   _cat=1;
   output;
   _trt=2;
   _ANYAE=1;
   _catord=1;
   _cat=1;
   output;
run;

*-------------------------------------------------------------------;
* Merge the _PCT dataset with its frameup dataset(_FRAME) ;
*-------------------------------------------------------------------;

proc sort data=_frame1;
   by _datasrt _blcksrt _cat _ANYAE _trt;
run;

proc sort data=_pct1;
```

```
  by _datasrt _blcksrt _cat _ANYAE _trt;
run;

data _pct1;
  merge _frame1(in=_inframe) _pct1;
  by _datasrt _blcksrt _cat _ANYAE _trt;

  if _inframe;

  if count=. then
    count=0;
run;

*-------------------------------------------------------------------;
* Delete Zero filled MISSING category rows for each combination of;
* _datasrt &_byvar _blcksrt;
*-------------------------------------------------------------------;

proc sort data=_pct1;
  by _datasrt _blcksrt _ANYAE;
run;

data _miss1(keep=_datasrt _blcksrt _ANYAE totcount);
  set _pct1;
  where _ANYAE=9998;
  retain totcount;
  by _datasrt _blcksrt _ANYAE;

  if first._ANYAE then
    totcount=0;
  totcount=totcount+count;

  if last._ANYAE;
run;

data _pct1(drop=totcount);
  merge _pct1 _miss1;
  by _datasrt _blcksrt _ANYAE;

  if totcount=0 then
    delete;
run;

*******************************************************************;
*IF PCTDISP=CAT/DPTVAR then add dptvar into denomitor frame dataset;
*******************************************************************;
*-------------------------------------------------------------------;
* Merge the _DENOMIN with its frame up dataset (_denomf) ;
*-------------------------------------------------------------------;

proc sort data=_denomf1;
  by _datasrt _cat;
run;
```

```
proc sort data=_denomin1;
  by _datasrt _cat;
run;

data _denomin1;
  merge _denomf1(in=_inframe) _denomin1;
  by _datasrt _cat;

  if _inframe;
  _blcksrt=1;
run;

*------------------------------------------------------------------;
* Merge in _PCT(counts) with the _DENOMIN(denominator for percents) ;
*------------------------------------------------------------------;

proc sort data=_pct1;
  by _datasrt _cat;
run;

*------------------------------------------------------------------;
* Create _VARNAME variable to hold depend variable name. ;
* Create _VRLABEL variable to display Group label. ;
* Create _RWLABEL variable to display &dptvar categories. ;
*------------------------------------------------------------------;

data _pct1;
  merge _denomin1(in=_a) _pct1;
  by _datasrt _cat;

  if _a;
  _varname="_ANYAE ";
  _vrlabel=" ";
  _rwlabel="Any event ";

  if _ANYAE=9998 then
    do;
      _rwlabel="Missing ";
      _catord=9998;
    end;
  else if _ANYAE=9999 then
    do;
      _rwlabel="Total ";
      _catord=9999;
    end;

  if _catord=. then
    _catord=9997;
run;

proc sort data=_pct1;
  by _datasrt _blcksrt _catord _ANYAE _trt _cat;
run;
```

```
*------------------------------------------------------------------;
* Create _CVALUE variable to display results. ;
* Create _ROWSRT variable to order results. ;
*------------------------------------------------------------------;

data _base1;
  length _catlabl $200 _cvalue $30.;
  set _pct1 end=eof;
  by _datasrt _blcksrt _catord _ANYAE _trt _cat;
  retain _rowsrt 0 _rowmax 0;
  array _trtcnt(*) _trt1-_trt3;
  drop _rowmax _cpct;
  length _cpct $100;
  _cpct=' ';
  _module='mcatstat';

  if count > . then
    _cvalue=put(count, 5.);
  else
    _cvalue=put(0, 5.);
  *------------------------------------------------------------------;
  * Format percent to append to display value in _CVALUE ;
  *------------------------------------------------------------------;

  if _trt ne . then
    do;

      if _trtcnt(_trt) > 0 then
        do;
          percent=count / _trtcnt(_trt) * 100;

          if percent > 0 then
            do;

              if round(percent, 0.1) GE 0.1 then
                _cpct="(*ESC*){nbspace 1}("||strip(put(percent, 5.1))||")";
              else
                _cpct="(*ESC*){nbspace 1}(0.0)";
              _cvalue=trim(_cvalue)||_cpct;
            end;
        end;
    end;

/*    if length(_cvalue) < 13 then */
/*      do; */
      *------------------------------------------------------------------;
      * Put character A0x at right most character to pad text;
      *------------------------------------------------------------------;
/*        substr(_cvalue, 13, 1)='A0'x; */
/*      end; */

  if first._ANYAE then
    do;
      _rowsrt=_rowsrt + 1;
      _rowmax=max(_rowsrt, _rowmax);
    end;
  _datatyp='data';
  _indent=0;
  _dptindt=0;
  _vorder=1;
  _rowjump=1;

  if upcase(_rwlabel)='_NONE_' then
    _rwlabel=' ';
  _indent=0;
  _dptindt=0;

  if _trt=2 +1 then
    _trt=9999;

  if eof then
    call symput('_rowsrt', compress(put(_rowmax, 4.)));
  _direct="TOP ";
  _p=2;
run;

********************************************************************************;
*SPECIFICATION 5.1 - *;
* 1) if user requested to calculate Observed proportion using Clopper & *;
* Pearson method then call macro cnp_obsprop. *;
********************************************************************************;
***hup02 modified to correct CI;

data _cnp _tmp_cnp;
  set _base1;

  if count=. then
    count=0;
  indc=1;
  output _cnp;
  indc=2;
  count=input(compress(symget('_trt'||compress(put(_trt, 8.)))), 8.) - count;
  output _cnp;

  if indc=2 and count=0 then
    output _tmp_cnp;
run;

proc sort data=_cnp;
  by _cat _rowsrt _rwlabel _trt;
run;

proc sort nodupkey data=_tmp_cnp(keep=_cat _rowsrt _rwlabel _trt);
  by _cat _rowsrt _rwlabel _trt;
run;

proc freq data=_cnp noprint;
  by _cat _rowsrt _rwlabel _trt;
  table indc/binomial alpha=0.05;
  output out=obsprop binomial;
  weight count;
run;

data obsprop;
  merge obsprop _tmp_cnp(in=a);
  by _cat _rowsrt _rwlabel _trt;

  if _bin_=1 and not a then
    do;
      xl_bin_=1 - xu_bin;
      xu_bin_=1 - xl_bin;
    end;
  else
    do;
      xl_bin_=xl_bin;
      xu_bin_=xu_bin;
    end;
run;

data cnpobsprop1(keep=_cat _rowsrt _rwlabel _trt cnp_ci);
  set obsprop;
  by _cat _rowsrt _rwlabel _trt;
  cnp_ci='(' || compress(put(xl_bin_ * 100, 5.1))
    || ',(*ESC*){nbspace 1}' || compress(put(xu_bin_ * 100, 5.1)) || ')';
  label cnp_ci='95% CI';
run;

proc datasets lib=work nolist gennum=all;
  delete _cnp obsprop;
  run;
  ;

proc sort data=_base1;
  by _cat _rowsrt _rwlabel _trt;
run;

proc sort data=cnpobsprop1;
  by _cat _rowsrt _rwlabel _trt;
run;

data _base1;
  length _varname $8 _vrlabel $200 _direct $20 _module $8 _rwlabel $800;
  ;
  merge _base1(in=a) cnpobsprop1;
  by _cat _rowsrt _rwlabel _trt;

  if a;
run;

******************************************************************************;
*SPECIFICATION 6 -1) G_A_TIERVAR - counts and percent *;
******************************************************************************;
```

```
proc transpose data=_data1 name=name out=lvlvars;
  var AEBODSYS AEDECOD;
run;

data _dsin;
  set _data1;
  output;
run;

data dsin;
  set _dsin;
  _cat=1;
  output;
run;

data frame;
  set _bydat1(keep=);
  _cat=1;
  _trt=1;
  output;
  _cat=1;
  _trt=2;
  output;
run;

proc sort data=frame;
  by _cat _trt;
run;

***************************************************************************************;
* SPECIFICATION 4 -1) denominator dataset and denominator variables *;
* 2) TRTN is N by treatment *;
* 3) SUBGRPN is N by treatment, by variables & subgroup or *;
* N by treatment and subgroup when PCTDISP=SGRPN *;
***************************************************************************************;

data denom(drop=_col) temp;
  _trt=1;
  trtn=input(symget(cats('_trt', _trt)), best.);
  _cat=1;
  subgrpn=trtn;
  output denom;
  _col=_trt;
  output temp;
  _trt=2;
  trtn=input(symget(cats('_trt', _trt)), best.);
  _cat=1;
  subgrpn=trtn;
  output denom;
  _col=_trt;
  output temp;
run;
```

```
*****************************************************************************;
* SPECIFICATION 5 -1) create _COL that reflect report column number *;
* 2) Call %util_num_obs and if no. of obs <=0 then exit *;
* out of macro. *;
* 3) reinitialize _rowsrt and _blcksrt based on joinyn *;
*****************************************************************************;

proc sql noprint;
  create table numin as select a.*, b._col from dsin as a left join temp as b on
    a._trt=b._trt;
quit;

data numin;
  set numin;
  where aeterm is not missing;
run;

*****************************************************************************;
* SPECIFICATION 6 -1) compute n by tier variables *;
* 2) setup sorting either by ALPHA/FREQ based on dptsort *;
*****************************************************************************;

data _null_;
  set lvlvars end=eof;
  length cclass cclasst $200;
  retain cclass cclasst;
  cclass=catx(' ', cclass, name);
  cclasst=catx(' ', cclasst, cats(name, '_n'));
  call symputx(cats('_cclass', _n_), cclass);

  if eof then
    do;
      call symputx('_cclasst', cclasst);
    end;
run;

proc sort data=numin out=class1(keep=AEBODSYS) nodupkey;
  by AEBODSYS;
run;

data classo1;
  set frame;

  do i=1 to cobs;
    set class1 nobs=cobs point=i;
    output;
  end;
run;

proc sort data=numin out=level1 NODUPKEY;
  by _cat _trt AEBODSYS usubjid;
run;

proc summary data=level1 nway classdata=classo1 missing;
```

```
  class _cat _trt AEBODSYS;
  output out=count1(drop=_type_ rename=(_freq_=count));
run;

data sort1;
  set class1;
  AEBODSYS_n=_n_;
run;

proc sort data=count1;
  by AEBODSYS;
run;

proc sort data=sort1 out=sorto1(keep=AEBODSYS AEBODSYS_n);
  by AEBODSYS;
run;

data countf1;
  length _rwlabel $800;
  merge count1 sorto1;
  by AEBODSYS;
  _level=1;
/*    _rwlabel=ifc(1>1, repeat(byte(160), 0)||AEBODSYS, AEBODSYS); */
  _rwlabel=ifc(1>1, repeat('~{nbspace 1}', 0)||AEBODSYS, AEBODSYS);
run;

proc sort data=numin out=class2(keep=AEBODSYS AEDECOD) nodupkey;
  by AEBODSYS AEDECOD;
run;

data classo2;
  set frame;

  do i=1 to cobs;
    set class2 nobs=cobs point=i;
    output;
  end;
run;

proc sort data=numin out=level2 NODUPKEY;
  by _cat _trt AEBODSYS AEDECOD usubjid;
run;

proc summary data=level2 nway classdata=classo2 missing;
  class _cat _trt AEBODSYS AEDECOD;
  output out=count2(drop=_type_ rename=(_freq_=count));
run;

proc sort data=level2 out=flevel2 nodupkey;
  by AEBODSYS AEDECOD usubjid;
  ;
run;

proc summary data=flevel2 nway classdata=class2 missing;
```

```
  class AEBODSYS AEDECOD;
  output out=fcount2(drop=_type_ rename=(_freq_=count));
run;

proc sort data=fcount2 out=freq2(keep=AEBODSYS AEDECOD count);
  by descending count AEBODSYS AEDECOD;
run;

data sort2;
  set freq2;
  AEDECOD_n=_n_;
run;

proc sort data=count2;
  by AEBODSYS AEDECOD;
run;

proc sort data=sort2 out=sorto2(keep=AEBODSYS AEDECOD AEDECOD_n);
  by AEBODSYS AEDECOD;
run;

data countf2;
  length _rwlabel $800;
  merge count2 sorto2;
  by AEBODSYS AEDECOD;
  _level=2;
/*    _rwlabel=ifc(2>1, repeat(byte(160), 5)||AEDECOD, AEDECOD); */
  _rwlabel=ifc(2>1, repeat('~{nbspace 1}', 5)||AEDECOD, AEDECOD);
run;

proc sort data=countf2;
  by AEBODSYS;
run;

data countf2;
  merge countf2 sorto1;
  by AEBODSYS;
run;

data alln;
  set countf1 countf2;
run;

*******************************************************************************;
* 1) gender specific denominator *;
* 2) compute percents *;
*******************************************************************************;

proc sort data=denom out=denomf(keep=_trt trtn) nodupkey;
  by _trt;
run;

proc sort data=alln out=alln;
  by _trt;
```

```
run;

data final;
  merge alln(in=a) denomf;
  by _trt;

  if a;
  length _cvalue $30;

  if count > . then
    _cvalue=put(count, 5.);
  else
    _cvalue=put(0, 5.);

  if count > 0 and trtn > 0 then
    do;
      pct=(count/trtn)*100;
      _cvalue=trim(_cvalue)||"  ("||strip(put(pct, 5.1))||")";
    end;

/*    if length(_cvalue) < 13 then */
/*      do; */
      *---------------------------------------------------------------;
      * Put character A0x at right most character to pad text;
      *---------------------------------------------------------------;
/*        substr(_cvalue, 13, 1)='A0'x; */
/*      end; */

  if _trt=3 then
    _trt=9999;
  _p=2;
run;

proc sort data=final;
  by _cat _trt AEBODSYS_n AEDECOD_n;
run;

data _base2;
  set final;
  by _cat _trt AEBODSYS_n AEDECOD_n;
  length _varname $8 _vrlabel $200 _direct $20 _module $8;

  if first._trt then
    _rowsrt=0;
  _rowsrt++1;
  _varname=" ";
  _vrlabel="";
  _vorder=1;
  _indent=coalesce(0, 0);
  _dptindt=coalesce(0, 0);
  _direct="TOP";
  _rowjump=1;
  _datatyp='data';
  _module="MCATSTAT";
  _datasrt=1;
  _blcksrt=1+ AEBODSYS_n;

  if _indent+_dptindt > 0 then
/*      _rwlabel=cats(repeat(byte(160), sum(_indent, _dptindt, 1)), _rwlabel); */
      _rwlabel=cats(repeat('~{nbspace 1}', sum(_indent, _dptindt, 1)), _rwlabel);
run;

proc sort data=_base2;
  by _datasrt _blcksrt _rowsrt;
run;

proc sort data=_base2 out=_dsnin1;
  by _rowsrt _rwlabel;
  where ^missing(_rwlabel);
run;

data _dsnin1 (keep=_rowsrt _rwlabel);
  set _dsnin1;
  by _rowsrt _rwlabel;

  if last._rwlabel;
run;

proc sort data=_dsnin1 out=_dsninx nodupkey dupout=_dupevent;
  by _rwlabel;
run;

proc datasets library=work;
  delete _dupevent _dsninx _dsnin1;
quit;

****************************************************************************************;
*SPECIFICATION 6.1 - *;
* 1) if user requested to calculate Observed proportion using Clopper & *;
* Pearson method then call macro cnp_obsprop. *;
****************************************************************************************;
***hup02 modified to correct CI;

data _cnp _tmp_cnp;
  set _base2;

  if count=. then
    count=0;
  indc=1;
  output _cnp;
  indc=2;
  count=input(compress(symget('_trt'||compress(put(_trt, 8.)))), 8.) - count;
  output _cnp;

  if indc=2 and count=0 then
    output _tmp_cnp;
run;
```

```
proc sort data=_cnp;
  by _cat AEBODSYS AEDECOD _rowsrt _rwlabel _trt;
run;

proc sort nodupkey data=_tmp_cnp(keep=_cat AEBODSYS AEDECOD _rowsrt _rwlabel
    _trt);
  by _cat AEBODSYS AEDECOD _rowsrt _rwlabel _trt;
run;

proc freq data=_cnp noprint;
  by _cat AEBODSYS AEDECOD _rowsrt _rwlabel _trt;
  table indc/binomial alpha=0.05;
  output out=obsprop binomial;
  weight count;
run;

data obsprop;
  merge obsprop _tmp_cnp(in=a);
  by _cat AEBODSYS AEDECOD _rowsrt _rwlabel _trt;

  if _bin_=1 and not a then
    do;
      xl_bin_=1 - xu_bin;
      xu_bin_=1 - xl_bin;
    end;
  else
    do;
      xl_bin_=xl_bin;
      xu_bin_=xu_bin;
    end;
run;

data cnpobsprop2(keep=_cat AEBODSYS AEDECOD _rowsrt _rwlabel _trt cnp_ci);
  set obsprop;
  by _cat AEBODSYS AEDECOD _rowsrt _rwlabel _trt;
  cnp_ci='(' || compress(put(xl_bin_ * 100, 5.1))
    || ',(*ESC*){nbspace 1}' || compress(put(xu_bin_ * 100, 5.1)) || ')';
  label cnp_ci='95% CI';
run;

proc datasets lib=work nolist gennum=all;
  delete _cnp obsprop;
  run;
  ;

proc sort data=_base2;
  by _cat AEBODSYS AEDECOD _rowsrt _rwlabel _trt;
run;

proc sort data=cnpobsprop2;
  by _cat AEBODSYS AEDECOD _rowsrt _rwlabel _trt;
run;

data _base2;
```

```
  merge _base2(in=a) cnpobsprop2;
  by _cat AEBODSYS AEDECOD _rowsrt _rwlabel _trt;

  if a;
run;

******************************************************************************;
*SPECIFICATION 12 -1) titles and footnotes *;
* 2) display *;
******************************************************************************;

data _final;
  set _base1 _base2;
run;

proc sort data=_final;
  by _datasrt _blcksrt _rowsrt;
run;

data ADAE_S130_SAE_UNB2_PED6;
  length _rvalue $200;
  set _final end=eof;
  _rwindt=sum(_indent, _dptindt);

  if _rwindt <=0 then
    _rvalue=_rwlabel;
  else
/*       _rvalue=repeat(byte(160), _rwindt-1)||_rwlabel; */
       _rvalue=repeat('~{nbspace 1}', _rwindt-1)||_rwlabel;
  _dummy=1;

  if _trt=. then
    _trt=1;
run;

proc sort data=ADAE_S130_SAE_UNB2_PED6;
  by _datasrt _trt _blcksrt _rowsrt;
run;

data treat;
  length FMTNAME $8 start 8 label $200;
  fmtname='TREAT';

  do start=1 to 2 + ("N"="Y");
    label=symget('_TRTLB'|| compress(put(start, 4.)));
    label=trim(label)
      || "|  (N(*ESC*){super a}=" || compress(symget("_TRT" || compress(put(start,
      4.)))) || ")";
    output;
  end;
run;

proc format cntlin=treat;
run;
```

```
data report;
  set ADAE_S130_SAE_UNB2_PED6;

  if _trt=9999 then
    _trt=2 +1;
  _rvalue=tranwrd(_rvalue, "|", '036e'x);
run;

proc sort data=report;
  by _datasrt _blcksrt _rowsrt _rvalue _trt;
run;

data data_1 (keep=_datasrt _blcksrt _rowsrt _rvalue COL:);
  set report;
  where _trt=1;
  rename _cvalue=COL11 cnp_ci=COL12;
run;

data data_2 (keep=_datasrt _blcksrt _rowsrt _rvalue COL:);
  set report;
  where _trt=2;
  rename _cvalue=COL21 cnp_ci=COL22;
run;

proc sort data=report out=extradata (keep=_datasrt _blcksrt _rowsrt _rvalue)
    nodupkey;
  by _datasrt _blcksrt _rowsrt _rvalue;
run;

data report;
  merge data_1 data_2 extradata;
  by _datasrt _blcksrt _rowsrt _rvalue;
run;

data report;
  set report;
  _fixvar=1;
  _fix2var=1;
  _dummy=1;
run;

proc sort data=report out=outdata1;
  by _datasrt _blcksrt _rowsrt _rvalue;
run;

ods escapechar="~";
title1 "Number (%) of Subjects Reporting at Least 1 New Serious Adverse Event After the EUA Snapshot, From Dose 1
to ";
title2 "Unblinding Date, by System Organ Class and Preferred Term (*ESC*){unicode 2013} Blinded Placebo-
Controlled Follow-up Period (*ESC*){unicode 2013}";
title3 "Phase 2/3 Subjects 12 Through 15 Years of Age (*ESC*){unicode 2013} Safety Population";
footnote1 "Abbreviation: EUA = emergency use authorization.";
footnote2 "Note: MedDRA (v24.0) coding dictionary applied.";
```

```
footnote3 "Note: Adverse events that occurred on the day of or after subjects were unblinded are excluded from this
summary.";
footnote4 "a.~{nbspace 5}N = number of subjects in the specified group, subjects who withdrew from the study before
EUA snapshot 25Mar2021 with the cutoff date 13Mar2021 are not included. This value is the denominator for the
percentage calculations.";
footnote5 "b.~{nbspace 5}n = Number of subjects reporting at least 1 occurrence of the specified event.  For (*ESC*)
{unicode 0022}any event,(*ESC*){unicode 0022} n = number of subjects reporting at least 1 occurrence of any event.";
footnote6
  "c.~{nbspace 5}Exact 2-sided CI based on the Clopper and Pearson method.";
ods html file="&outtable.";

proc report data=outdata1 nowd list missing contents="" split="|"
    style(report)={} style(header)={} style(column)={};
  column _fixvar _fix2var _datasrt _blcksrt _rowsrt ("" _rvalue) (" " (" " ("Vaccine Group (as Administered)~{line}"
("BNT162b2 (30 (*ESC*){unicode 03BC}g)| (N(*ESC*){super a}=&_TRT1.)" (COL11
    COL12)) ("Placebo| (N(*ESC*){super a}=&_TRT2.)" (COL21 COL22))) ) ) _dummy;
  define _fixvar / group noprint;
  define _fix2var / group noprint;
  define _datasrt / group order=internal noprint;
  define _blcksrt / group order=internal noprint;
  define _rowsrt / group order=internal noprint;
  define _rvalue / group "System Organ Class|~{nbspace 5}Preferred Term"
    order=data style(column)={just=left width=60mm} style(header)={just=left}
    left;
  ;
  define COL11 / group nozero " n(*ESC*){super b} (%)"
    style(column)={leftmargin=12px} style(header)={just=center} center;
  define COL12 / group nozero " (95%~{nbspace 1}CI(*ESC*){super c})"
    style(column)={leftmargin=12px} style(header)={just=center} center;
  define COL21 / group nozero " n(*ESC*){super b} (%)"
    style(column)={leftmargin=12px} style(header)={just=center} center;
  define COL22 / group nozero " (95%~{nbspace 1}CI(*ESC*){super c})"
    style(column)={leftmargin=12px} style(header)={just=center} center;
  define _dummy / sum noprint;
  break before _fixvar / contents="" page;
  compute before _fix2var;
    line @1 " ~n ";
  endcomp;
  compute after _blcksrt;
    line " ~n ";
  endcomp;
run;
```