

```

*****
** Program Name   : adsl_fu_d2_ped_saf.sas                **
** Date Created  : 10Mar2021                            **
** Programmer Name : (b) (4), (b) (6)                   **
** Purpose       : Create adsl_fu_d2_ped_saf            **
** Input data    : adsl                                  **
** Output file   : adsl_fu_d2_ped_saf.html              **
*****
options mprint mlogic symbolgen mprint symbolgen mlogic nocenter missing=" ";
ods escapechar="~";

proc datasets library=WORK kill nolist nodetails;
quit;

**Setup the environment**
%let prot=/Volumes/app/cdars/prod/sites/cdars4/prjC459/nda2_unblinded_esub/euaext_esub_adam/saseng/cdisc3_0;
libname datvprot "&prot./data_vai" access=readonly;
%let outpath=&prot./analysis/esub;
%let outlog=&outpath./logs/adsl_fu_d2_ped_saf.log;
%let outtable=&outpath./output/adsl_fu_d2_ped_saf.html;
*****
* Clean *;
*****
options mprint mlogic symbolgen;
title;
footnote;

proc delete data=work._all_;
run;

proc printto log="&outlog" new;
run;

data adsl;
  set DATVPROT.ADSL;
  length FUP2CAT1_FUP2CAT2_ $ 100;
  FUP2CATN=min(FUP2CA1N, FUP2CA2N);

  if FUP2CATN=FUP2CA1N then
    FUP2CAT=FUP2CAT1;

  if FUP2CATN=FUP2CA2N then
    FUP2CAT=FUP2CAT2;
  FUP2CAT1_=FUP2CAT;

  if FUP2CATN=1 then
    do;
      FUP2CAT1_="(*ESC*){unicode 003c}1 Month";
      FUP2CA1N_=1;
    end;
  else if 3 >=FUP2CATN>1 then
    do;
      FUP2CAT1_="(*ESC*){unicode 2265}"||tranwrd(FUP2CAT1_,"-",

```

```

        " Months to (*ESC*){unicode 003c}");
    FUP2CA1N_=FUP2CATN;
end;

if 3<FUP2CATN then
    do;
        FUP2CA1N_=4;
        FUP2CAT1_="(*ESC*){unicode 2265}3 Months";
    end;
    FUP2CAT1_=tranwrd(FUP2CAT1_, "1 Months", "1 Month");
run;

data g_adsl_dsin;
    set adsl;
    where agegr4n=1 and saffl="Y" and MULENRFL ne "Y" and phasen ne 1;
run;

data __trtmap;
    length trtcode trtdec $100;

    if 0 then
        set g_adsl_dsin(keep=TRT01AN);
        trtval=1;

    if vtype(TRT01AN)='C' then
        trtcode=tranwrd(compbl(quote("8")), ' ', "" );
    else
        trtcode="8";
        trtdec="BNT162b2 (30 (*ESC*){unicode 03BC}g)";
        trtvar="TRT01AN";
        trtlbl="TRT01A";
        output;
        trtval=2;

    if vtype(TRT01AN)='C' then
        trtcode=tranwrd(compbl(quote("9")), ' ', "" );
    else
        trtcode="9";
        trtdec="Placebo";
        trtvar="TRT01AN";
        trtlbl="TRT01A";
        output;
        trtval=3;

    if vtype(TRT01AN)='C' then
        trtcode=tranwrd(compbl(quote("8 9")), ' ', "" );
    else
        trtcode="8 9";
        trtdec="Total";
        trtvar="TRT01AN";
        trtlbl="TRT01A";
        output;
        stop;
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;

  if TRT01AN in (8 9) then
    do;
      newtrtn=3;
      newtrt=coalescec("Total", TRT01A);
      output;
    end;
run;

```

```

data _subGrpData(compress=no);
  delete;
run;

```

```

*-----;
* Initialize dataset for non-pvalue footnote queue. ;
*-----;

```

```

data _stdft1(compress=no);
  length model $200 mark $5;
  index=0;
  model=' ';
  mark=' ';
run;

```

```

*-----;
* Initialize dataset for pvalue related footnote queue.;
*-----;

```

```

data _stdft2(compress=no);
  length model $200 mark $5;
  index=0;
  model=' ';
  mark=' ';
run;

```

```

*-----;
* Initialize structure for _BASETEMPLATE dataset. ;

```

```

*-----;

data _basetemplate(compress=no);
  length _varname $8 _cvalue $35 _direct $20 _vrlabel $200 _rwlabel
    _colabel $800 _datatyp $5 _module $8 _pr_lbl $ 200;
  array _c _character_;
  delete;
run;

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

*-----;
* Count number of treatment groups ;
*-----;

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

*-----;
* Generate variable _TRT. Use assigned order if applicable ;
*-----;

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;

      *-----;
      * Generate _STR as the treatment label ;
      *-----;
      _str=NEWTRT;
      *-----;

```

```
* Update _TRTLB&n with generated treatment label ;
*-----;
```

```
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 - :_trt3 from (select distinct
    USUBJID, _trt from _data1 where NEWTRTN is not missing) group by _trt;
```

```
select compress(put(count(*), 5.)) into :_trt4 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;
```

```
data _bydat1;
```

```
set _bydat1 end=eof;
```

```
by _datasrt;
```

```
retain _preby 0;
```

```
drop _preby;
```

```
_byvar1=0;
```

```
if eof then
```

```
do;
```

```
call symput("_preby1", compress(put(_byvar1, 4.)));
```

```
if 0=0 then
```

```
output;
```

```
end;
```

```
run;
```

```
data _bydat1;
```

```
set _bydat1;
```

```
by _datasrt;
```

```
length _bycol _byindnt $50 _bylast $10;
```

```
_bycol=" ";
```

```
_byindnt=" ";
```

```
_bylast=" ";
```

```
run;
```

```
proc sort data=_bydat1;
```

```
by _datasrt;
```

```

run;

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

data _null_;
  set _data1 end=eof;

  if eof then
    call symput('dptlab', vlabel(FUP2CA1N_));
run;

data _anal1;
  length FUP2CA1N_ 8;
  set _data1;
  where same and FUP2CA1N_ is not missing;
  _blcksrt=1;
  _cnt=1;
  _cat=1;

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

proc sort data=_anal1;
  by _datasrt _blcksrt FUP2CA1N_ _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 FUP2CA1N_ _trt USUBJID;
run;

proc freq data=_temp91;
  format FUP2CA1N_;
  tables _datasrt*_blcksrt*_cat * FUP2CA1N_ * _trt / sparse norow nocol
    nopercnt 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;
  _trt=3;
count=&_trt3;
output;
run;

data _denomf1;
  _datasrt=1;
set _bydat1(keep=);
* All treatment groups ;
  _trt1=0;
  _trt2=0;
  _trt3=0;
* _CAT is the subgroup variable ;
  _cat=1;
output;
run;

proc transpose data=_denomf1 out=_denomin1(drop=_name__label_) prefix=_trt;
  by _datasrt _cat;
  var count;
  id _trt;
run;

proc sql noprint;
  select put(nobs - delobs, 12.) into :_nobs from dictionary.tables
         where (libname="WORK" and memname="_PCT1");
  select setting into :miss from dictionary.options where
         upcase(optname)="MISSING";
quit;

proc sort data=_pct1 out=_expv1 (keep=_datasrt _blcksrt FUP2CA1N_) nodupkey;
  by _datasrt _blcksrt FUP2CA1N_;
run;

proc sort data=_expv1;
  by _datasrt _blcksrt FUP2CA1N_;
run;

proc sort data=_anall out=_catlabel1 (keep=_datasrt _blcksrt FUP2CA1N_
  FUP2CAT1_) nodupkey;
  by _datasrt _blcksrt FUP2CA1N_;
  ;
run;

data _expv1;
  merge _expv1 (in=_a) _catlabel1 (in=_b);
  by _datasrt _blcksrt FUP2CA1N_;

```

```

    if _a;
run;

proc sql noprint;
    select put(nobs - delobs, 12.) into :_nobs from dictionary.tables
        where (libname="WORK" and memname="_PCT1");
    select setting into :miss from dictionary.options where
        upcase(optname)="MISSING";
quit;

```

```

proc sort data=_expv1;
    by _datasrt _blcksrt FUP2CA1N_;
run;

```

```

data _frame1;
    set _expv1;
    by _datasrt _blcksrt FUP2CA1N_;
    length _catLabl $100;
    _catLabl=FUP2CAT1_;

```

```

    if first._blcksrt then
        _catord=0;
    _catord + 1;
    _trt=1;
    _cat=1;
output;
    _trt=2;
    _cat=1;
output;
    _trt=3;
    _cat=1;
output;

```

```

run;

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

```

```

proc sort data=_frame1;
    by _datasrt _blcksrt _cat FUP2CA1N__trt;
run;

```

```

proc sort data=_pct1;
    by _datasrt _blcksrt _cat FUP2CA1N__trt;
run;

```

```

data _pct1;
    merge _frame1(in=_inframe) _pct1;
    by _datasrt _blcksrt _cat FUP2CA1N__trt;

    if _inframe;

    if count=. then

```



```

        count=0;
run;

proc sort data=_pct1;
    by _datasrt _blcksrt FUP2CA1N_;
run;

data _miss1(keep=_datasrt _blcksrt FUP2CA1N_ totcount);
    set _pct1;
    where FUP2CA1N_=9998;
    retain totcount;
    by _datasrt _blcksrt FUP2CA1N_;

    if first.FUP2CA1N_ then
        totcount=0;
    totcount=totcount+count;

    if last.FUP2CA1N_;
run;

data _pct1(drop=totcount);
    merge _pct1 _miss1;
    by _datasrt _blcksrt FUP2CA1N_;

    if totcount=0 then
        delete;
run;

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;

data _pct1;
    if 0 then

```

```

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

if _a;
  _varname="FUP2CA1N_ ";
  _vrlabel="Total exposure from Dose 2 to cutoff date ";
  _rwlabel=_catLabel;

if FUP2CA1N_=9998 then
  do;
    _rwlabel="Missing ";
    _catord=9998;
  end;
else if FUP2CA1N_=9999 then
  do;
    _rwlabel="Total ";
    _catord=9999;
  end;

if _catord=. then
  _catord=9997;
run;

proc sort data=_pct1;
  by _datasrt _blcksrt _catord FUP2CA1N__trt _cat;
run;

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

data _base1;
  length _catlabl $200;
  set _pct1 end=eof;
  by _datasrt _blcksrt _catord FUP2CA1N__trt _cat;
  retain _rowsrt 0 _rowmax 0;
  array _trcnt(*) _trt1-_trt4;
  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*){nbspspace 1}("||strip(put(percent, 5.1))||")";
                    else
                        _cpct="(*ESC*){nbspspace 1}(0.0)";
                        _cvalue=trim(_cvalue)||_cpct;
                    end;
                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.FUP2CA1N_ then
    do;
        _rowsrt=_rowsrt + 1;
        _rowmax=max(_rowsrt, _rowmax);
    end;
    _datatype='data';
    _indent=0;
    _dptindt=0;
    _vorder=1;
    _rowjump=1;

if upcase(_rwlable)=' _NONE_' then
    _rwlable=' ';
    _indent=4;
    _dptindt=0;

if _trt=3 +1 then
    _trt=9999;

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

data _base1;
set _base1;
    _vhlabel="Subjects (%)" with length of follow-up of:";
run;

```

```

ods escapechar="~";
title1 "Follow-up Time After Dose 2 (*ESC*){unicode 2013} Subjects 12 Through 15 Years of Age (*ESC*){unicode 2013} Safety Population";
footnote1 "Note: Follow-up time was calculated to the cutoff date or the date of unblinding, whichever date was earlier.";
footnote2 "a.(*ESC*){nbsp 5}N = number of subjects in the specified group, or the total sample. This value is the denominator for the percentage calculations.";
footnote3
    "b.(*ESC*){nbsp 5}n = Number of subjects with the specified characteristic.";

data _final;
    set _base1;
run;

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

*-----;
* At least one of TRT and STAT is vertical;
*-----;

data _final;
    set _final;
    drop __trt;

    if __trt=9999 then
        __trt=3 + 1;
    else
        __trt=__trt;

    if __trt=. then
        __trt=1;
        _column=__trt;

    if _column=9999 then
        _column=3 + 1;
run;

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

data _linecnt;
    set _final end=eof;
    by _datasrt _blcksrt _rowsrt _column;
    retain _totline _maxval _maxrow _rwlbttag _vrlbttag 0 _maxline _linecnt;
    keep _datasrt _blcksrt _totline _linecnt _maxrow;

    if _rowjump=. then
        _rowjump=1;

    if first._blcksrt then

```

```

do;
  *-----;
  * Count words inside DATA step ;
  *-----;
  _token=repeat(' ', 99);
  _count=1;
  _token=scan(_vrlabel, _count, "|");

  if _token=: ' ' then
    _tag=1;
  else
    _tag=0;

  do while(_token ^=' ');
    _count=_count + 1;
    _token=scan(_vrlabel, _count, "|");
  end;
  _linecnt=_count - 1 + _tag;
  ;
  _totline=_linecnt;

  if _vrlabel ne '' and _vrlabel ne '^' & _datatype='data' then
    _vrlbtag=1;
end;

if first._rowsrt then
do;
  *-----;
  * Count words inside DATA step ;
  *-----;
  _token=repeat(' ', 99);
  _count=1;
  _token=scan(_rwlabel, _count, "|");

  if _token=: ' ' then
    _tag=1;
  else
    _tag=0;

  do while(_token ^=' ');
    _maxrow=max(_maxrow, length(_token) + _indent);
    _count=_count + 1;
    _token=scan(_rwlabel, _count, "|");
  end;
  _maxline=_count - 1 + _tag;
  ;

  if _rwlabel ne '' then
    _rwlbttag=1;
    _totline + _rowjump - 1;
end;
*-----;
* Count words inside DATA step ;
*-----;

```

```

_token=repeat(' ', 99);
_count=1;
_token=scan(_cvalue, _count, "|");

if _token=: ' ' then
    _tag=1;
else
    _tag=0;

do while(_token ^=' ');
    _maxval=max(_maxval, length(_token));
    _count=_count + 1;
    _token=scan(_cvalue, _count, "|");
end;
_ccnt=_count - 1 + _tag;
;
_maxline=max(_maxline, _ccnt);

if last._rowsrt then
    _totline=_maxline + _totline;

if last._blcksrt then
    do;
        _totline=_totline - _rowjump + 1;
        output;
    end;

if eof then
    do;
        call symput('_valwid', compress(put(_maxval, 3.)));
        call symput('_rwlbtg', put(_rwlbtg, 1.));
        call symput('_vrlbtg', put(_vrlbtg, 1.));
    end;

run;

data _final;
length _direct $20;
_direct=' ';
merge _final_linecnt;
by _datasrt _blcksrt;

run;

proc sql noprint;
create table rson as select distinct _trt, _column , _vrlabel as _rwlable ,
    _datasrt, _blcksrt, (min(_rowsrt)-0.5) as _rowsrt , _dptindt as _indent , 0
as _dptindt from _final(where=( _vrlabel^=' ')) group by _trt, _column ,
    _datasrt, _blcksrt, _vrlabel;

quit;

proc sql noprint;
create table hson as select distinct _trt, _column , _vhlable as _rwlable ,
    _datasrt, _blcksrt, (min(_rowsrt)-0.9) as _rowsrt , _dptindt as _indent , 0
as _dptindt from _final(where=( _vhlable^=' ')) group by _trt, _column ,
    _datasrt, _blcksrt, _vhlable;

```

```

quit;

data ADSL_FU_D2_PED_SAF;
  length _rvalue $800;
  set _final rspan hspan end=eof;
  _rwindt=sum(_indent, _dptindt);

  if _rwindt <=0 then
    _rvalue=_rlabel;
  else
    _rvalue=repeat(byte(160), _rwindt-1)||_rlabel;
  _dummy=1;

  if _trt=. then
    _trt=1;
run;

proc sort data=ADSL_FU_D2_PED_SAF;
  by _datasrt _trt _blcksrt _rowsrt;
run;

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

  do start=1 to 3 + ("N"="Y");
    label=symget('_TRTLB'|| compress(put(start, 4.)));
    label=trim(label)
      || " (N~{super a}=" || compress(symget("_TRT" || compress(put(start,
      4.)))) || ")"
    || "|n~{super b} (%)";
    output;
  end;
run;

proc format cntlin=treat;
run;

data outdata1;
  set ADSL_FU_D2_PED_SAF;

  if upcase(_module)='MCATSTAT' then
    _cvalue=transtrn(compress(_cvalue), '(', ' ');
  _fixvar=1;
  _fix2var=1;
run;

option nobyline;

proc sort data=outdata1;
  by _datasrt _trt _blcksrt _rowsrt;
run;

proc sql noprint;

```

```

select distinct start, label into :start1, :_trlb11 - :_trlb199 from treat
    order by start;
quit;

proc sort data=outdata1 out=_pre_transposed;
    by _fixvar _fix2var _datasrt _blcksrt _rowsrt _rvalue _trt;
run;

data _pre_transposed;
    set _pre_transposed;

    if _trt=9999 then
        _trt=3 +1;
run;

proc transpose data=_pre_transposed out=_column_transposed (drop=_name_)
    prefix=TRT;
    by _fixvar _fix2var _datasrt _blcksrt _rowsrt _rvalue;
    var _cvalue;
    id _trt;
run;

data REPORT;
    set _column_transposed;
    _dummy=1;
run;

proc sort data=report;
    by _datasrt _blcksrt _rowsrt _dummy;
run;

ods html file="&prot./analysis/esub/output/adsl_fu_d2_ped_saf.html";

proc report data=report nowd list missing contents="" split=""
    style(report)={} style(header)={} style(column)={};
    column _fixvar _fix2var _datasrt _blcksrt _rowsrt (" " _rvalue)
        (("Vaccine Group (as Administered)~{line}" TRT1 TRT2) TRT3 _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 " " order=data style(column)={just=left width=60mm
        rightmargin=18px} style(header)={just=left} left;
    define _dummy / sum noprint;
    define TRT1 / group nozero "&_trlb11." spacing=2 style(column)={width=35mm
        leftmargin=12px} style(header)={just=center} center;
    define TRT2 / group nozero "&_trlb12." spacing=2 style(column)={width=35mm
        leftmargin=12px} style(header)={just=center} center;
    define TRT3 / group nozero "&_trlb13." spacing=2 style(column)={width=35mm
        leftmargin=12px} style(header)={just=center} center;
    break before _fixvar / contents="" page;
    compute before _fix2var;
    line @1 " ~n ";

```



```
endcomp;  
compute after _blcksrt;  
  line " ~n ";  
endcomp;  
run;  
  
ods HTML close;  
proc printto;  
run;
```