

```

*****
** Program Name   : adds-s002-ped-rand.sas                **,
** Date Created  : 21Mar2021                             **,
** Programmer Name : (b) (6)                             **,
** Purpose       : Create adds-s002-ped-rand              **,
** Input data    : adds adsl                             **,
** Output data   : adds-s002-ped-rand.html                **,
*****
%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 codename=adds-s002-ped-rand;
%let outlog=&prot./analysis/esub/logs/&codename..log;
%let outtable=&prot./analysis/esub/output/&codename..html;

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

options mprint mlogic symbolgen mprint symbolgen mlogic nocenter missing=" ";
title;
footnote;

proc datasets library=WORK kill nolist nodetails;
quit;

/* Format */
proc format;
    value dsdecod 1="Adverse event" 3="Death" 5="Lost to follow-up" 7="Other"
        8="Physician decision" 9="Pregnancy" 11="Protocol deviation"
        13="Screen Failure" 14="Study terminated by sponsor"
        16="Withdrawal by subject"
        17="Medication error without associated adverse event"
        18="No longer meets eligibility criteria"
        25="Refused further study procedures" 26="Withdrawal by parent/guardian";
run;

/* Readin ADSL */
data ds;
    set datvprot.adds;
run;

data compltdt;
    set datvprot.adds;

    if dsdecodn=2 and dsphase='VACCINATION' and astdt ne .;
    keep usubjid astdt;
run;

proc sql undo_policy=none;
    create table ds as select a.*, b.astdt as compltdt from ds a left join
        compltdt b on a.usubjid=b.usubjid;
quit;

```

```

proc sort data=ds;
    by usubjid;
run;

data adds;
    merge ds(in=a) datvprot.adsl(in=b);
    by usubjid;

    if a;
run;

data g_a_dsin;
    set adds;
    analysis_subset='Y';
    where dsphase ne 'OPEN LABEL TREATMENT' and M1P2EXC ne 'Y';
run;

data g_adsl_dsin;
    set datvprot.adsl;
    where RANDFL eq 'Y' and phasen not in (1) and AGEGR4N ne . and MULENRFL ne "Y";
run;

proc sql noprint;
    create table a_dsin as select * from g_a_dsin where usubjid in (select
        distinct usubjid from g_adsl_dsin);
quit;

data __trtmap;
    length trtcode trtdec $100;

    if 0 then
        set g_adsl_dsin(keep=TRT01PN);
    trtval=1;

    if vtype(TRT01PN)='C' then
        trtcode=tranwrd(compbl(quote("8")), ' ', "" "");
    else
        trtcode="8";
    trtdec="BNT162b2 (30 (*ESC*){unicode 03BC}g)~{line}";
    trtvar="TRT01PN";
    trtlbl="TRT01P";
    output;
    ;
    ;
    ;
    trtval=2;

    if vtype(TRT01PN)='C' then
        trtcode=tranwrd(compbl(quote("9")), ' ', "" "");
    else
        trtcode="9";
    trtdec="Placebo~{line}";
    trtvar="TRT01PN";
    trtlbl="TRT01P";

```

```

output;
stop;
run;

data g_adsl_dsin;
set g_adsl_dsin;

if TRT01PN in (8) then
do;
newtrtn=1;
newtrt=coalescec("BNT162b2 (30 (*ESC*){unicode 03BC}g)~{line}", TRT01P);
output;
end;

if TRT01PN in (9) then
do;
newtrtn=2;
newtrt=coalescec("Placebo~{line}", TRT01P);
output;
end;

run;

data g_a_dsin;
set g_a_dsin;

if TRT01PN in (8) then
do;
newtrtn=1;
newtrt=coalescec("BNT162b2 (30 (*ESC*){unicode 03BC}g)~{line}", TRT01P);
output;
end;

if TRT01PN in (9) then
do;
newtrtn=2;
newtrt=coalescec("Placebo~{line}", TRT01P);
output;
end;

run;

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

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

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

if d1;
run;

```

```

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 final;
    where (NEWTRTN is not missing);
run;

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;

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;

* Handle sub-group N=xxx/sub-group analysis request ;

proc sql noprint;
    select count(unique AGEGR4) into : _subGrpN from _data1 where AGEGR4 is not
        missing;

```

```

create table _subGrpData as select distinct _trt, AGEGR4, count(distinct
    USUBJID) as _subGrpCnt from _data1 where AGEGR4 is not missing group by _trt,
    AGEGR4;
quit;

proc sql noprint;
create table _subGrpDataVH as select distinct 9999 as _trt, AGEGR4,
    count(distinct USUBJID) as _subGrpCnt from _data1 where AGEGR4 is not missing
    group by AGEGR4;
quit;

data _subGrpData;
length _cat $100;
set _subGrpData;
by _trt;
_cat=AGEGR4;
run;

data _trtframe;
_trt=ifN(1 eq 3, 9999, 1);
output;
_trt=ifN(2 eq 3, 9999, 2);
output;
run;

proc sql noprint;
create table _fullSubGrp as select * from (select distinct _trt from
    _trtframe), (select distinct _cat from _subGrpData) order by _trt, _cat;
quit;

data _fullSubGrp;
set _fullSubGrp;
by _trt _cat;

if first._trt then
    _subcat=0;
    _subcat + 1;

if _subcat=2 + 1 then
    _subcat=9999;
run;

data _subGrpData _tmpsubGrpData;
merge _fullSubGrp _subGrpData;
by _trt _cat;
length _colabel $200;
_colabel=_cat;

if _subGrpCnt=. then
    _subGrpCnt=0;
run;

data _subGrpData;
set _subGrpData;

```

```

        _colabel=trim(_colabel) || "(N~{super a}=" || compress(put(_subGrpCnt, 5.))
        || ")";
run;

data _subGrpDataVH;
    length _cat $100;
    set _subGrpDataVH;
    by _trt;
    _cat=AGEGR4;
run;

proc sql noprint;
    create table _fullSubGrpVH as select * from (select distinct _trt from
        _subGrpDataVH), (select distinct _cat from _subgrpdatavh) order by _trt, _cat;

data _fullSubGrpVH;
    set _fullSubGrpVH;
    by _trt _cat;

    if first._trt then
        _subcat=0;
        _subcat + 1;

    if _subcat=2 + 1 then
        _subcat=9999;
run;

data _subGrpDataVH _tmpsubGrpDataVH;
    merge _subGrpDataVH _fullSubGrpVH;
    by _trt _cat;
    length _colabel $200;
    _colabel=_cat;

    if _subGrpCnt=. then
        _subGrpCnt=0;
run;

data _subGrpDataVH;
    set _subGrpDataVH;
    _colabel=trim(_colabel) || "(N~{super a}=" || compress(put(_subGrpCnt, 5.))
    || ")";
run;

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;

/* Create criteria flags */
data _data1;
    set _data1;
    _event1=ifN(RANDFL in ('Y'), 1, 0);
    _event2=ifN(RANDFL eq 'Y' and (VAX101DT eq . and VAX102DT eq .), 1, 0);
    _event3=ifN(RANDFL eq 'Y' and VAX101DT ne ., 1, 0);
    _event4=ifN(RANDFL eq 'Y' and VAX102DT ne ., 1, 0);
    _event5=ifN(RANDFL eq 'Y' and DSPHASEN=26 and dsdecodn=2, 1, 0);
    _event6=ifN(RANDFL eq 'Y' and DSPHASEN=26 and EOTDCDT ne . and (EOSDCDT
        eq . or EOSDCDT>M1P2CUT>.) and dsdecodn not in (. 2) and (VAX101DT ne . or
        VAX102DT ne .), 1, 0);
    _event7=ifN(RANDFL eq 'Y' and DSPHASEN=26 and EOTDCDT ne . and (EOSDCDT
        eq . or EOSDCDT>M1P2CUT>.) and dsdecodn not in (. 2) and vax101dt
        ne . and (vax102dt eq . or astdt < vax102dt), 1, 0);
    _event8=ifN(RANDFL eq 'Y' and DSPHASEN=26 and EOTDCDT ne . and (EOSDCDT
        eq . or EOSDCDT>M1P2CUT>.) and dsdecodn not in (. 2) and vax101dt ne . and
        vax102dt ne . and (vax102dt <=astdt and (M1PD2DT eq . or astdt<M1PD2DT)), 1,
        0);
    _event10=ifN(RANDFL eq 'Y' and DSPHASEN=31 and EOSDCDT ne . and dsdecodn not
        in (. 2) and (VAX101DT ne . or VAX102DT ne .) and COMPLTDT ne EOSDCDT, 1, 0);
    _event11=ifN(RANDFL eq 'Y' and DSPHASEN=31 and EOSDCDT ne . and dsdecodn not
        in (. 2) and vax101dt ne . and (vax102dt eq . or astdt < vax102dt), 1, 0);
    _event12=ifN(RANDFL eq 'Y' and DSPHASEN=31 and EOSDCDT ne . and dsdecodn not
        in (. 2) and vax101dt ne . and vax102dt ne . and (vax102dt <=astdt
        and (M1PD2DT eq . or astdt<M1PD2DT)), 1, 0);
run;

/* Crit 1 */

```

```

data _anall;
  length _EVENT1 8;
  length _cat $100;
  set _data1;
  where AGEGR4 is not missing;
  where same and _EVENT1 is not missing;
  _blcksrt=1;
  _cnt=1;
  _cat=AGEGR4;

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

proc sort data=_anall;
  by _datasrt _blcksrt _EVENT1 _trt _cat;
run;

data _subgrpvar;
  set _data1 (keep=AGEGR4);
  where ^missing(AGEGR4);
  format AGEGR4;
run;

proc sql noprint;
  select distinct AGEGR4 into :subv1 - : subv2 from _subgrpvar where AGEGR4 is
    not missing;
quit;

data _templ;
  set _anall;
  output;
run;

proc sort data=_templ out=_temp91 nodupkey;
  by _datasrt _blcksrt _cat _EVENT1 _trt usubjid;
  ;
run;

proc freq data=_temp91;
  format _EVENT1;
  tables _datasrt*_blcksrt*_cat * _EVENT1 * _trt / sparse norow nocol nopercnt
    out=_pct1(drop=percent);
run;

proc sort data=_templ out=_analcnt1 nodupkey;
  by _datasrt _cat _trt USUBJID;
  where RANDFL eq 'Y';
run;

proc freq data=_analcnt1 noprint;
  tables _datasrt*_cat * _trt / sparse noprint out=_denom1(drop=percent);
run;

```



```

data _denomf1;
    length _cat $100;
    _datasrt=1;
    set _bydat1(keep=);
    * All treatment groups ;
    _trt1=0;
    _trt2=0;
    * _CAT is the subgroup variable ;
    _cat="&subv1.";
    output;
    _cat="&subv2.";
    output;
run;

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

data _frame1;
    _datasrt=1;
    set _bydat1(keep=);
    _blcksrt=1;
    length _EVENT1 8;
    length _cat $100;
    _catLabl=" ";
    _trt=1;
    _EVENT1=1;
    _catord=1;
    _subcat=1;
    _cat="&subv1.";
    output;
    _subcat=2;
    _cat="&subv2.";
    output;
    _trt=2;
    _EVENT1=1;
    _catord=1;
    _subcat=1;
    _cat="&subv1.";
    output;
    _subcat=2;
    _cat="&subv2.";
    output;
run;

proc sort data=_frame1;
    by _datasrt _blcksrt _cat _EVENT1 _trt;
run;

proc sort data=_pct1;
    by _datasrt _blcksrt _cat _EVENT1 _trt;

```

```

run;

data _pct1;
  merge _frame1(in=_inframe) _pct1;
  by _datasrt _blcksrt _cat _EVENT1 _trt;

  if _inframe;

  if count=. then
    count=0;

run;

proc sort data=_pct1;
  by _datasrt _blcksrt _EVENT1;
run;

data _miss1(keep=_datasrt _blcksrt _EVENT1 totcount);
  set _pct1;
  where _EVENT1=9998;
  retain totcount;
  by _datasrt _blcksrt _EVENT1;

  if first._EVENT1 then
    totcount=0;
  totcount=totcount+count;

  if last._EVENT1;

run;

data _pct1(drop=totcount);
  merge _pct1 _miss1;
  by _datasrt _blcksrt _EVENT1;

  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;

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="_EVENT1 ";
    _vrlabel=" ";
    _rwlabel="Randomized ";

    if _EVENT1=9998 then
        do;
            _rwlabel="Missing ";
            _catord=9998;
        end;
    else if _EVENT1=9999 then
        do;
            _rwlabel="Total ";
            _catord=9999;
        end;

    if _catord=. then
        _catord=9997;
run;

proc sort data=_pct1;
    by _datasrt _blcksrt _catord _EVENT1 _trt _cat;
run;

data _base1;
    length _catlabl $200;
    set _pct1 end=eof;
    by _datasrt _blcksrt _catord _EVENT1 _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.);

    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;

```

```

if length(_cvalue) < 13 then
    do;
        substr(_cvalue, 13, 1)='A0'x;
    end;

```

```

if first._EVENT1 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;
```

```

/* Crit 2 */
data _anal2;
    length _EVENT2 8;
    length _cat $100;
    set _data1;
    where AGEGR4 is not missing;
    where same and _EVENT2 is not missing;
    _blcksrt=1;
    _cnt=1;
    _cat=AGEGR4;

```

```

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

proc sort data=_anal2;
    by _datasrt _blcksrt _EVENT2 _trt _cat;
run;

data _subgrpvar;
    set _data1 (keep=AGEGR4);
    where ^missing(AGEGR4);
    format AGEGR4;
run;

proc sql noprint;
    select distinct AGEGR4 into :subv1 - : subv2 from _subgrpvar where AGEGR4 is
        not missing;
quit;

data _temp2;
    set _anal2;
    output;
run;

proc sort data=_temp2 out=_temp92 nodupkey;
    by _datasrt _blcksrt _cat _EVENT2 _trt usubjid;
    ;
run;

proc freq data=_temp92;
    format _EVENT2;
    tables _datasrt*_blcksrt*_cat * _EVENT2 * _trt / sparse norow nocol nopercnt
        out=_pct2(drop=percent);
run;

proc sort data=_temp2 out=_analcnt2 nodupkey;
    by _datasrt _cat _trt USUBJID;
    where RANDFL eq 'Y';
run;

proc freq data=_analcnt2 noprint;
    tables _datasrt*_cat * _trt / sparse noprint out=_denom2(drop=percent);
run;

data _denomf2;
    length _cat $100;
    _datasrt=1;
    set _bydat1(keep=);
    * All treatment groups ;
    _trt1=0;
    _trt2=0;
    * _CAT is the subgroup variable ;

```

```

    _cat("&subv1.");
output;
    _cat("&subv2.");
output;
run;

proc transpose data=_denom2 out=_denomin2(drop=_name__label_) prefix=_trt;
    by _datasrt _cat;
    var count;
    id _trt;
run;

data _frame2;
    _datasrt=1;
    set _bydat1(keep=);
    _blcksrt=1;
    length _EVENT2 8;
    length _cat $100;
    _catLabl=" ";
    _trt=1;
    _EVENT2=1;
    _catord=1;
    _subcat=1;
    _cat("&subv1.");
output;
    _subcat=2;
    _cat("&subv2.");
output;
    _trt=2;
    _EVENT2=1;
    _catord=1;
    _subcat=1;
    _cat("&subv1.");
output;
    _subcat=2;
    _cat("&subv2.");
output;
run;

proc sort data=_frame2;
    by _datasrt _blcksrt _cat _EVENT2 _trt;
run;

proc sort data=_pct2;
    by _datasrt _blcksrt _cat _EVENT2 _trt;
run;

data _pct2;
    merge _frame2(in=_inframe) _pct2;
    by _datasrt _blcksrt _cat _EVENT2 _trt;

    if _inframe;

    if count=. then

```

```

        count=0;
run;

proc sort data=_pct2;
    by _datasrt _blcksrt _EVENT2;
run;

data _miss2(keep=_datasrt _blcksrt _EVENT2 totcount);
    set _pct2;
    where _EVENT2=9998;
    retain totcount;
    by _datasrt _blcksrt _EVENT2;

    if first._EVENT2 then
        totcount=0;
    totcount=totcount+count;

    if last._EVENT2;
run;

data _pct2(drop=totcount);
    merge _pct2 _miss2;
    by _datasrt _blcksrt _EVENT2;

    if totcount=0 then
        delete;
run;

proc sort data=_denomf2;
    by _datasrt _cat;
run;

proc sort data=_denomin2;
    by _datasrt _cat;
run;

data _denomin2;
    merge _denomf2(in=_inframe) _denomin2;
    by _datasrt _cat;

    if _inframe;
    _blcksrt=1;
run;

proc sort data=_pct2;
    by _datasrt _cat;
run;

data _pct2;
    if 0 then
        set _basetemplate;
    merge _denomin2(in=_a) _pct2;
    by _datasrt _cat;

```

```

if _a;
  _varname="_EVENT2 ";
  _vrlabel=" ";
  _rwlabel="Not vaccinated ";

if _EVENT2=9998 then
  do;
    _rwlabel="Missing ";
    _catord=9998;
  end;
else if _EVENT2=9999 then
  do;
    _rwlabel="Total ";
    _catord=9999;
  end;

if _catord=. then
  _catord=9997;

run;

proc sort data=_pct2;
  by _datasrt _blcksrt _catord _EVENT2 _trt _cat;
run;

data _base2;
  length _catlabl $200;
  set _pct2 end=eof;
  by _datasrt _blcksrt _catord _EVENT2 _trt _cat;
  retain _rowsrt 1 _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.);

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)";
          end;
        end;
      end;
    end;
  end;

```



```

        _cvalue=trim(_cvalue)||_cpct;
    end;
end;

if length(_cvalue) < 13 then
    do;
        substr(_cvalue, 13, 1)='A0'x;
    end;

if first._EVENT2 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;

/* Crit 3 */
data _anal3;
    length _EVENT3 8;
    length _cat $100;
    set _data1;
    where AGEGR4 is not missing;
    where same and _EVENT3 is not missing;
    _blcksrt=1;
    _cnt=1;
    _cat=AGEGR4;

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

proc sort data=_anal3;
    by _datasrt _blcksrt _EVENT3 _trt _cat;
run;

```

```

data _subgrpvar;
  set _data1 (keep=AGEGR4);
  where ^missing(AGEGR4);
  format AGEGR4;
run;

proc sql noprint;
  select distinct AGEGR4 into :subv1 - : subv2 from _subgrpvar where AGEGR4 is
    not missing;
quit;

data _temp3;
  set _anal3;
  output;
run;

proc sort data=_temp3 out=_temp93 nodupkey;
  by _datasrt _blcksrt _cat _EVENT3 _trt usubjid;
  ;
run;

proc freq data=_temp93;
  format _EVENT3;
  tables _datasrt*_blcksrt*_cat * _EVENT3 * _trt / sparse norow nocol nopercnt
    out=_pct3(drop=percent);
run;

proc sort data=_temp3 out=_analcnt3 nodupkey;
  by _datasrt _cat _trt USUBJID;
  where RANDFL eq 'Y';
run;

proc freq data=_analcnt3 noprint;
  tables _datasrt*_cat * _trt / sparse noprint out=_denom3(drop=percent);
run;

data _denomf3;
  length _cat $100;
  _datasrt=1;
  set _bydat1(keep=);
  * All treatment groups ;
  _trt1=0;
  _trt2=0;
  * _CAT is the subgroup variable ;
  _cat="&subv1.";
  output;
  _cat="&subv2.";
  output;
run;

proc transpose data=_denom3 out=_denomin3(drop=_name _label_) prefix=_trt;
  by _datasrt _cat;
  var count;

```

```

        id_trt;
run;

data _frame3;
    _datasrt=1;
    set _bydat1(keep=);
    _blcksrt=1;
    length _EVENT3 8;
    length _cat $100;
    _catLabl=" ";
    _trt=1;
    _EVENT3=1;
    _catord=1;
    _subcat=1;
    _cat="&subv1.";
output;
    _subcat=2;
    _cat="&subv2.";
output;
    _trt=2;
    _EVENT3=1;
    _catord=1;
    _subcat=1;
    _cat="&subv1.";
output;
    _subcat=2;
    _cat="&subv2.";
output;
run;

proc sort data=_frame3;
    by _datasrt _blcksrt _cat _EVENT3 _trt;
run;

proc sort data=_pct3;
    by _datasrt _blcksrt _cat _EVENT3 _trt;
run;

data _pct3;
    merge _frame3(in=_inframe) _pct3;
    by _datasrt _blcksrt _cat _EVENT3 _trt;

    if _inframe;

    if count=. then
        count=0;
run;

proc sort data=_pct3;
    by _datasrt _blcksrt _EVENT3;
run;

data _miss3(keep=_datasrt _blcksrt _EVENT3 totcount);
    set _pct3;

```

```

where _EVENT3=9998;
retain totcount;
by _datasrt _blcksrt _EVENT3;

if first._EVENT3 then
    totcount=0;
totcount=totcount+count;

if last._EVENT3;
run;

data _pct3(drop=totcount);
merge _pct3 _miss3;
by _datasrt _blcksrt _EVENT3;

if totcount=0 then
    delete;
run;

proc sort data=_denomf3;
by _datasrt _cat;
run;

proc sort data=_denomin3;
by _datasrt _cat;
run;

data _denomin3;
merge _denomf3(in=_inframe) _denomin3;
by _datasrt _cat;

if _inframe;
    _blcksrt=1;
run;

proc sort data=_pct3;
by _datasrt _cat;
run;

data _pct3;
if 0 then
    set _basemplate;
merge _denomin3(in=_a) _pct3;
by _datasrt _cat;

if _a;
    _varname="_EVENT3 ";
    _vrlabel="Vaccinated ";
    _rwlabel="Dose 1 ";

if _EVENT3=9998 then
    do;
        _rwlabel="Missing ";
        _catord=9998;

```

```

        end;
    else if _EVENT3=9999 then
        do;
            _rwlabel="Total ";
            _catord=9999;
        end;

    if _catord=. then
        _catord=9997;

run;

proc sort data=_pct3;
    by _datasrt _blcksrt _catord _EVENT3 _trt _cat;
run;

data _base3;
    length _catlabl $200;
    set _pct3 end=eof;
    by _datasrt _blcksrt _catord _EVENT3 _trt _cat;
    retain _rowsrt 2 _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.);

    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;

        do;

            if length(_cvalue) < 13 then
                do;
                    substr(_cvalue, 13, 1)='A0'x;
                end;
            end;
        end;
    end;
end;

```

```

if first._EVENT3 then
  do;
    _rowsrt=_rowsrt + 1;
    _rowmax=max(_rowsrt, _rowmax);
  end;
  _datatype='data';
  _indent=0;
  _dptindt=0;
  _vorder=1;
  _rowjump=1;

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

  if _trt=2 +1 then
    _trt=9999;

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

```

```

/* Crit 4 */
data _anal4;
  length _EVENT4 8;
  length _cat $100;
  set _data1;
  where AGEGR4 is not missing;
  where same and _EVENT4 is not missing;
  _blcksrt=1;
  _cnt=1;
  _cat=AGEGR4;

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

```

```

proc sort data=_anal4;
  by _datasrt _blcksrt _EVENT4 _trt _cat;
run;

```

```

data _subgrpvar;
  set _data1 (keep=AGEGR4);
  where ^missing(AGEGR4);
  format AGEGR4;
run;

```

```

proc sql noprint;
  select distinct AGEGR4 into :subv1 - : subv2 from _subgrpvar where AGEGR4 is

```

FDA-CBER-2022-5812-0735885

```

        not missing;
quit;

data _temp4;
    set _anal4;
    output;
run;

proc sort data=_temp4 out=_temp94 nodupkey;
    by _datasrt _blcksrt _cat _EVENT4 _trt usubjid;
    ;
run;

proc freq data=_temp94;
    format _EVENT4;
    tables _datasrt*_blcksrt*_cat * _EVENT4 * _trt / sparse norow nocol nopercnt
        out=_pct4(drop=percent);
run;

proc sort data=_temp4 out=_analcnt4 nodupkey;
    by _datasrt _cat _trt USUBJID;
    where RANDFL eq 'Y';
run;

proc freq data=_analcnt4 noprint;
    tables _datasrt*_cat * _trt / sparse noprint out=_denom4(drop=percent);
run;

data _denomf4;
    length _cat $100;
    _datasrt=1;
    set _bydat1(keep=);
    * All treatment groups ;
    _trt1=0;
    _trt2=0;
    * _CAT is the subgroup variable ;
    _cat="&subv1.";
    output;
    _cat="&subv2.";
    output;
run;

proc transpose data=_denom4 out=_denomin4(drop=_name _label_) prefix=_trt;
    by _datasrt _cat;
    var count;
    id _trt;
run;

data _frame4;
    _datasrt=1;
    set _bydat1(keep=);
    _blcksrt=1;
    length _EVENT4 8;
    length _cat $100;

```

```

    _catLabl=" ";
    _trt=1;
    _EVENT4=1;
    _catord=1;
    _subcat=1;
    _cat="&subv1.";
output;
    _subcat=2;
    _cat="&subv2.";
output;
    _trt=2;
    _EVENT4=1;
    _catord=1;
    _subcat=1;
    _cat="&subv1.";
output;
    _subcat=2;
    _cat="&subv2.";
output;
run;

proc sort data=_frame4;
    by _datasrt _blcksrt _cat _EVENT4 _trt;
run;

proc sort data=_pct4;
    by _datasrt _blcksrt _cat _EVENT4 _trt;
run;

data _pct4;
    merge _frame4(in=_inframe) _pct4;
    by _datasrt _blcksrt _cat _EVENT4 _trt;

    if _inframe;

    if count=. then
        count=0;
run;

proc sort data=_pct4;
    by _datasrt _blcksrt _EVENT4;
run;

data _miss4(keep=_datasrt _blcksrt _EVENT4 totcount);
    set _pct4;
    where _EVENT4=9998;
    retain totcount;
    by _datasrt _blcksrt _EVENT4;

    if first._EVENT4 then
        totcount=0;
    totcount=totcount+count;

    if last._EVENT4;

```



```

run;

data _pct4(drop=totcount);
  merge _pct4 _miss4;
  by _datasrt _blcksrt _EVENT4;

  if totcount=0 then
    delete;
run;

proc sort data=_denomf4;
  by _datasrt _cat;
run;

proc sort data=_denomin4;
  by _datasrt _cat;
run;

data _denomin4;
  merge _denomf4(in=_inframe) _denomin4;
  by _datasrt _cat;

  if _inframe;
  _blcksrt=1;
run;

proc sort data=_pct4;
  by _datasrt _cat;
run;

data _pct4;
  if 0 then
    set _basetemplate;
  merge _denomin4(in=_a) _pct4;
  by _datasrt _cat;

  if _a;
  _varname="_EVENT4 ";
  _vrlabel=" ";
  _rwlabel="Dose 2 ";

  if _EVENT4=9998 then
    do;
      _rwlabel="Missing ";
      _catord=9998;
    end;
  else if _EVENT4=9999 then
    do;
      _rwlabel="Total ";
      _catord=9999;
    end;

  if _catord=. then
    _catord=9997;

```

```

run;

proc sort data=_pct4;
    by _datasrt _blcksrt _catord _EVENT4 _trt _cat;
run;

data _base4;
    length _catlabl $200;
    set _pct4 end=eof;
    by _datasrt _blcksrt _catord _EVENT4 _trt _cat;
    retain _rowsrt 3 _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.);

    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;

        do;

            if length(_cvalue) < 13 then
                do;
                    substr(_cvalue, 13, 1)='A0'x;
                end;

            if first._EVENT4 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=4;
_dptindt=0;

if _trt=2 +1 then
    _trt=9999;

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

/* Crit 5 */
data _anal5;
length _EVENT5 8;
length _cat $100;
set _data1;
where AGEGR4 is not missing;
where same and _EVENT5 is not missing;
_blcksrt=2;
_cnt=1;
_cat=AGEGR4;

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

proc sort data=_anal5;
by _datasrt _blcksrt _EVENT5 _trt _cat;
run;

data _subgrpvar;
set _data1 (keep=AGEGR4);
where ^missing(AGEGR4);
format AGEGR4;
run;

proc sql noprint;
select distinct AGEGR4 into :subv1 - : subv2 from _subgrpvar where AGEGR4 is
    not missing;
quit;

data _temp5;
set _anal5;
output;
run;

proc sort data=_temp5 out=_temp95 nodupkey;

```

```

    by _datasrt _blcksrt _cat _EVENT5 _trt usubjid;
    ;
run;

proc freq data=_temp95;
    format _EVENT5;
    tables _datasrt*_blcksrt*_cat * _EVENT5 * _trt / sparse norow nocol nopercnt
        out=_pct5(drop=percent);
run;

proc sort data=_temp5 out=_analcnt5 nodupkey;
    by _datasrt _cat _trt USUBJID;
    where RANDFL eq 'Y';
run;

proc freq data=_analcnt5 noprint;
    tables _datasrt*_cat * _trt / sparse noprint out=_denom5(drop=percent);
run;

data _denomf5;
    length _cat $100;
    _datasrt=1;
    set _bydat1(keep=);
    * All treatment groups ;
    _trt1=0;
    _trt2=0;
    * _CAT is the subgroup variable ;
    _cat="&subv1.";
    output;
    _cat="&subv2.";
    output;
run;

proc transpose data=_denom5 out=_denomin5(drop=_name _label_) prefix=_trt;
    by _datasrt _cat;
    var count;
    id _trt;
run;

data _frame5;
    _datasrt=1;
    set _bydat1(keep=);
    _blcksrt=2;
    length _EVENT5 8;
    length _cat $100;
    _catLabl=" ";
    _trt=1;
    _EVENT5=1;
    _catord=1;
    _subcat=1;
    _cat="&subv1.";
    output;
    _subcat=2;
    _cat="&subv2.";

```

```

output;
  _trt=2;
  _EVENT5=1;
  _catord=1;
  _subcat=1;
  _cat="&subv1.";
output;
  _subcat=2;
  _cat="&subv2.";
output;
run;

proc sort data=_frame5;
  by _datasrt _blcksrt _cat _EVENT5 _trt;
run;

proc sort data=_pct5;
  by _datasrt _blcksrt _cat _EVENT5 _trt;
run;

data _pct5;
  merge _frame5(in=_inframe) _pct5;
  by _datasrt _blcksrt _cat _EVENT5 _trt;

  if _inframe;

  if count=. then
    count=0;
run;

proc sort data=_pct5;
  by _datasrt _blcksrt _EVENT5;
run;

data _miss5(keep=_datasrt _blcksrt _EVENT5 totcount);
  set _pct5;
  where _EVENT5=9998;
  retain totcount;
  by _datasrt _blcksrt _EVENT5;

  if first._EVENT5 then
    totcount=0;
  totcount=totcount+count;

  if last._EVENT5;
run;

data _pct5(drop=totcount);
  merge _pct5 _miss5;
  by _datasrt _blcksrt _EVENT5;

  if totcount=0 then
    delete;
run;

```

```

proc sort data=_denomf5;
    by _datasrt _cat;
run;

proc sort data=_denomin5;
    by _datasrt _cat;
run;

data _denomin5;
    merge _denomf5(in=_inframe) _denomin5;
    by _datasrt _cat;

    if _inframe;
    _blcksrt=2;
run;

proc sort data=_pct5;
    by _datasrt _cat;
run;

data _pct5;
    if 0 then
        set _basetemplate;
    merge _denomin5(in=_a) _pct5;
    by _datasrt _cat;

    if _a;
    _varname="_EVENT5 ";
    _vrlabel=" ";
    _rwlabel="Completed 1-month post(*ESC*){unicode 2013}Dose 2 visit (vaccination period) ";

    if _EVENT5=9998 then
        do;
            _rwlabel="Missing ";
            _catord=9998;
        end;
    else if _EVENT5=9999 then
        do;
            _rwlabel="Total ";
            _catord=9999;
        end;

    if _catord=. then
        _catord=9997;
run;

proc sort data=_pct5;
    by _datasrt _blcksrt _catord _EVENT5 _trt _cat;
run;

data _base5;
    length _catlabl $200;
    set _pct5 end=eof;

```

```

by _datasrt _blcksrt _cator _EVENT5 _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.);

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;
        substr(_cvalue, 13, 1)='A0'x;
    end;

if first._EVENT5 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;

/* Crit 6 */
data _anal6;
    length _EVENT6 8;
    length _cat $100;
    set _data1;
    where AGEGR4 is not missing;
    where same and _EVENT6 is not missing;
    _blksrt=2;
    _cnt=1;
    _cat=AGEGR4;

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

proc sort data=_anal6;
    by _datasrt _blksrt _EVENT6 _trt _cat;
run;

data _subgrpvar;
    set _data1 (keep=AGEGR4);
    where ^missing(AGEGR4);
    format AGEGR4;
run;

proc sql noprint;
    select distinct AGEGR4 into :subv1 - : subv2 from _subgrpvar where AGEGR4 is
        not missing;
quit;

data _temp6;
    set _anal6;
    output;
run;

proc sort data=_temp6 out=_temp96 nodupkey;
    by _datasrt _blksrt _cat _EVENT6 _trt usubjid;
    ;
run;

proc freq data=_temp96;
    format _EVENT6;
    tables _datasrt*_blksrt*_cat * _EVENT6 * _trt / sparse norow nocol nopercnt
        out=_pct6(drop=percent);
run;

```



```

proc sort data=_temp6 out=_analcnt6 nodupkey;
  by _datasrt _cat _trt USUBJID;
  where RANDFL eq 'Y';
run;

proc freq data=_analcnt6 noprint;
  tables _datasrt*_cat * _trt / sparse noprint out=_denom6(drop=percent);
run;

data _denomf6;
  length _cat $100;
  _datasrt=1;
  set _bydat1(keep=);
  * All treatment groups ;
  _trt1=0;
  _trt2=0;
  * _CAT is the subgroup variable ;
  _cat="&subv1.";
  output;
  _cat="&subv2.";
  output;
run;

proc transpose data=_denom6 out=_denomin6(drop=_name __label_) prefix=_trt;
  by _datasrt _cat;
  var count;
  id _trt;
run;

data _frame6;
  _datasrt=1;
  set _bydat1(keep=);
  _blcksrt=2;
  length _EVENT6 8;
  length _cat $100;
  _catLabl=" ";
  _trt=1;
  _EVENT6=1;
  _catord=1;
  _subcat=1;
  _cat="&subv1.";
  output;
  _subcat=2;
  _cat="&subv2.";
  output;
  _trt=2;
  _EVENT6=1;
  _catord=1;
  _subcat=1;
  _cat="&subv1.";
  output;
  _subcat=2;
  _cat="&subv2.";

```

```

        output;
run;

proc sort data=_frame6;
    by _datasrt _blcksrt _cat _EVENT6 _trt;
run;

proc sort data=_pct6;
    by _datasrt _blcksrt _cat _EVENT6 _trt;
run;

data _pct6;
    merge _frame6(in=_inframe) _pct6;
    by _datasrt _blcksrt _cat _EVENT6 _trt;

    if _inframe;

    if count=. then
        count=0;
run;

proc sort data=_pct6;
    by _datasrt _blcksrt _EVENT6;
run;

data _miss6(keep=_datasrt _blcksrt _EVENT6 totcount);
    set _pct6;
    where _EVENT6=9998;
    retain totcount;
    by _datasrt _blcksrt _EVENT6;

    if first._EVENT6 then
        totcount=0;
    totcount=totcount+count;

    if last._EVENT6;
run;

data _pct6(drop=totcount);
    merge _pct6 _miss6;
    by _datasrt _blcksrt _EVENT6;

    if totcount=0 then
        delete;
run;

proc sort data=_denomf6;
    by _datasrt _cat;
run;

proc sort data=_denomin6;
    by _datasrt _cat;
run;

```

```

data _denomin6;
  merge _denomf6(in=_inframe) _denomin6;
  by _datasrt _cat;

  if _inframe;
  _blcksrt=2;
run;

proc sort data=_pct6;
  by _datasrt _cat;
run;

data _pct6;
  if 0 then
    set _basetemplate;
  merge _denomin6(in=_a) _pct6;
  by _datasrt _cat;

  if _a;
  _varname="_EVENT6 ";
  _vrlabel=" ";
  _rwlabel="Discontinued from vaccination period but continue in the study up to 1-month post(*ESC*){unicode
2013}Dose 2 visit ";

  if _EVENT6=9998 then
    do;
      _rwlabel="Missing ";
      _catord=9998;
    end;
  else if _EVENT6=9999 then
    do;
      _rwlabel="Total ";
      _catord=9999;
    end;

  if _catord=. then
    _catord=9997;
run;

proc sort data=_pct6;
  by _datasrt _blcksrt _catord _EVENT6 _trt _cat;
run;

data _base6;
  length _catlabl $200;
  set _pct6 end=eof;
  by _datasrt _blcksrt _catord _EVENT6 _trt _cat;
  retain _rowsrt 1 _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.);

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;
      substr(_cvalue, 13, 1)='A0'x;
    end;

  if first._EVENT6 then
    do;
      _rowsrt=_rowsrt + 1;
      _rowmax=max(_rowsrt, _rowmax);
    end;
  _datatype='data';
  _indent=0;
  _dptindt=0;
  _vorder=1;
  _rowjump=1;

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

  if _trt=2 +1 then
    _trt=9999;

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

run;

/* Crit 7 */

```

```

data _anal7;
    length _EVENT7 8;
    length _cat $100;
    set _data1;
    where AGEGR4 is not missing;
    where same and _EVENT7 is not missing;
    _blcksrt=2;
    _cnt=1;
    _cat=AGEGR4;

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

proc sort data=_anal7;
    by _datasrt _blcksrt _EVENT7 _trt _cat;
run;

data _subgrpvar;
    set _data1 (keep=AGEGR4);
    where ^missing(AGEGR4);
    format AGEGR4;
run;

proc sql noprint;
    select distinct AGEGR4 into :subv1 - : subv2 from _subgrpvar where AGEGR4 is
        not missing;
quit;

data _temp7;
    set _anal7;
    output;
run;

proc sort data=_temp7 out=_temp97 nodupkey;
    by _datasrt _blcksrt _cat _EVENT7 _trt usubjid;
    ;
run;

proc freq data=_temp97;
    format _EVENT7;
    tables _datasrt*_blcksrt*_cat * _EVENT7 * _trt / sparse norow nocol nopercnt
        out=_pct7(drop=percent);
run;

proc sort data=_temp7 out=_analcnt7 nodupkey;
    by _datasrt _cat _trt USUBJID;
    where RANDFL eq 'Y';
run;

proc freq data=_analcnt7 noprint;
    tables _datasrt*_cat * _trt / sparse noprint out=_denom7(drop=percent);
run;

```

```

data _denom7;
    length _cat $100;
    _datasrt=1;
    set _bydat1(keep=);
    * All treatment groups ;
    _trt1=0;
    _trt2=0;
    * _CAT is the subgroup variable ;
    _cat="&subv1.";
    output;
    _cat="&subv2.";
    output;
run;

proc transpose data=_denom7 out=_denomin7(drop=_name __label_) prefix=_trt;
    by _datasrt _cat;
    var count;
    id _trt;
run;

data _frame7;
    _datasrt=1;
    set _bydat1(keep=);
    _blcksrt=2;
    length _EVENT7 8;
    length _cat $100;
    _catLabl=" ";
    _trt=1;
    _EVENT7=1;
    _catord=1;
    _subcat=1;
    _cat="&subv1.";
    output;
    _subcat=2;
    _cat="&subv2.";
    output;
    _trt=2;
    _EVENT7=1;
    _catord=1;
    _subcat=1;
    _cat="&subv1.";
    output;
    _subcat=2;
    _cat="&subv2.";
    output;
run;

proc sort data=_frame7;
    by _datasrt _blcksrt _cat _EVENT7 _trt;
run;

proc sort data=_pct7;
    by _datasrt _blcksrt _cat _EVENT7 _trt;

```

```

run;

data _pct7;
  merge _frame7(in=_inframe) _pct7;
  by _datasrt _blcksrt _cat _EVENT7 _trt;

  if _inframe;

  if count=. then
    count=0;

run;

proc sort data=_pct7;
  by _datasrt _blcksrt _EVENT7;
run;

data _miss7(keep=_datasrt _blcksrt _EVENT7 totcount);
  set _pct7;
  where _EVENT7=9998;
  retain totcount;
  by _datasrt _blcksrt _EVENT7;

  if first._EVENT7 then
    totcount=0;
  totcount=totcount+count;

  if last._EVENT7;

run;

data _pct7(drop=totcount);
  merge _pct7 _miss7;
  by _datasrt _blcksrt _EVENT7;

  if totcount=0 then
    delete;

run;

proc sort data=_denomf7;
  by _datasrt _cat;
run;

proc sort data=_denomin7;
  by _datasrt _cat;
run;

data _denomin7;
  merge _denomf7(in=_inframe) _denomin7;
  by _datasrt _cat;

  if _inframe;
  _blcksrt=2;

run;

proc sort data=_pct7;

```

```

        by _datasrt _cat;
run;

data _pct7;
    if 0 then
        set _basetemplate;
    merge _denomin7(in=_a) _pct7;
    by _datasrt _cat;

    if _a;
    _varname="_EVENT7 ";
    _vrlabel=" ";
    _rwlabel="Discontinued after Dose 1 and before Dose 2 ";

    if _EVENT7=9998 then
        do;
            _rwlabel="Missing ";
            _catord=9998;
        end;
    else if _EVENT7=9999 then
        do;
            _rwlabel="Total ";
            _catord=9999;
        end;

    if _catord=. then
        _catord=9997;
run;

proc sort data=_pct7;
    by _datasrt _blcksrt _catord _EVENT7 _trt _cat;
run;

data _base7;
    length _catlabl $200;
    set _pct7 end=eof;
    by _datasrt _blcksrt _catord _EVENT7 _trt _cat;
    retain _rowsrt 2 _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.);

    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;

```

```

if length(_cvalue) < 13 then
    do;
        substr(_cvalue, 13, 1)='A0'x;
    end;

```

```

if first._EVENT7 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=4;
_dptindt=0;

```

```

if _trt=2 +1 then
    _trt=9999;

```

```

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

```

run;

```

/* Crit 8 */
data _anal8;
    length _EVENT8 8;
    length _cat $100;
    set _data1;
    where AGEGR4 is not missing;
    where same and _EVENT8 is not missing;
    _blcksrt=2;
    _cnt=1;
    _cat=AGEGR4;

```

```

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

proc sort data=_anal8;
    by _datasrt _blcksrt _EVENT8 _trt _cat;
run;

data _subgrpvar;
    set _data1 (keep=AGEGR4);
    where ^missing(AGEGR4);
    format AGEGR4;
run;

proc sql noprint;
    select distinct AGEGR4 into :subv1 - : subv2 from _subgrpvar where AGEGR4 is
        not missing;
quit;

data _temp8;
    set _anal8;
    output;
run;

proc sort data=_temp8 out=_temp98 nodupkey;
    by _datasrt _blcksrt _cat _EVENT8 _trt usubjid;
    ;
run;

proc freq data=_temp98;
    format _EVENT8;
    tables _datasrt*_blcksrt*_cat * _EVENT8 * _trt / sparse norow nocol nopercnt
        out=_pct8(drop=percent);
run;

proc sort data=_temp8 out=_analcnt8 nodupkey;
    by _datasrt _cat _trt USUBJID;
    where RANDFL eq 'Y';
run;

proc freq data=_analcnt8 noprint;
    tables _datasrt*_cat * _trt / sparse noprint out=_denom8(drop=percent);
run;

data _denomf8;
    length _cat $100;
    _datasrt=1;
    set _bydat1(keep=);
    * All treatment groups ;
    _trt1=0;
    _trt2=0;
    * _CAT is the subgroup variable ;

```

```

    _cat("&subv1.");
output;
    _cat("&subv2.");
output;
run;

proc transpose data=_denom8 out=_denomin8(drop=_name__label_) prefix=_trt;
    by _datasrt _cat;
    var count;
    id _trt;
run;

data _frame8;
    _datasrt=1;
    set _bydat1(keep=);
    _blcksrt=2;
    length _EVENT8 8;
    length _cat $100;
    _catLabl=" ";
    _trt=1;
    _EVENT8=1;
    _catord=1;
    _subcat=1;
    _cat("&subv1.");
output;
    _subcat=2;
    _cat("&subv2.");
output;
    _trt=2;
    _EVENT8=1;
    _catord=1;
    _subcat=1;
    _cat("&subv1.");
output;
    _subcat=2;
    _cat("&subv2.");
output;
run;

proc sort data=_frame8;
    by _datasrt _blcksrt _cat _EVENT8 _trt;
run;

proc sort data=_pct8;
    by _datasrt _blcksrt _cat _EVENT8 _trt;
run;

data _pct8;
    merge _frame8(in=_inframe) _pct8;
    by _datasrt _blcksrt _cat _EVENT8 _trt;

    if _inframe;

    if count=. then

```

```

        count=0;
run;

proc sort data=_pct8;
    by _datasrt _blcksrt _EVENT8;
run;

data _miss8(keep=_datasrt _blcksrt _EVENT8 totcount);
    set _pct8;
    where _EVENT8=9998;
    retain totcount;
    by _datasrt _blcksrt _EVENT8;

    if first._EVENT8 then
        totcount=0;
    totcount=totcount+count;

    if last._EVENT8;
run;

data _pct8(drop=totcount);
    merge _pct8 _miss8;
    by _datasrt _blcksrt _EVENT8;

    if totcount=0 then
        delete;
run;

proc sort data=_denomf8;
    by _datasrt _cat;
run;

proc sort data=_denomin8;
    by _datasrt _cat;
run;

data _denomin8;
    merge _denomf8(in=_inframe) _denomin8;
    by _datasrt _cat;

    if _inframe;
    _blcksrt=2;
run;

proc sort data=_pct8;
    by _datasrt _cat;
run;

data _pct8;
    if 0 then
        set _basetemplate;
    merge _denomin8(in=_a) _pct8;
    by _datasrt _cat;

```

```

if _a;
  _varname="_EVENT8 ";
  _vrlabel=" ";
  _rwlabel="Discontinued after Dose 2 and before 1-month post(*ESC*){unicode 2013}Dose 2 visit ";

if _EVENT8=9998 then
  do;
    _rwlabel="Missing ";
    _catord=9998;
  end;
else if _EVENT8=9999 then
  do;
    _rwlabel="Total ";
    _catord=9999;
  end;

if _catord=. then
  _catord=9997;

run;

proc sort data=_pct8;
  by _datasrt _blcksrt _catord _EVENT8 _trt _cat;
run;

data _base8;
  length _catlabl $200;
  set _pct8 end=eof;
  by _datasrt _blcksrt _catord _EVENT8 _trt _cat;
  retain _rowsrt 3 _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.);

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)";
          end;
        end;
      end;
  end;

```

```

        _cvalue=trim(_cvalue)||_cpct;
    end;
end;

if length(_cvalue) < 13 then
    do;
        substr(_cvalue, 13, 1)='A0'x;
    end;

if first._EVENT8 then
    do;
        _rowsrt=_rowsrt + 1;
        _rowmax=max(_rowsrt, _rowmax);
    end;
    _datatyp='data';
    _indent=0;
    _dptindt=0;
    _vorder=1;
    _rowjump=1;

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

if _trt=2 +1 then
    _trt=9999;

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

/* Crit 9 */
data _anal9;
    length DSDECODN 8;
    length _cat $100;
    set _data1;
    where AGEGR4 is not missing;
    where same and DSDECODN is not missing;
    _blksrt=2;
    _cnt=1;
    _cat=AGEGR4;

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

proc sort data=_anal9;
    by _datasrt _blksrt DSDECODN _trt _cat;
run;

```

```

data _subgrpvar;
    set _data1 (keep=AGEGR4);
    where ^missing(AGEGR4);
    format AGEGR4;
run;

proc sql noprint;
    select distinct AGEGR4 into :subv1 - : subv2 from _subgrpvar where AGEGR4 is
        not missing;
quit;

data _temp9;
    set _anal9;
    output;
run;

proc sort data=_temp9 out=_temp99 nodupkey;
    by _datasrt _blcksrt _cat DSDECODN _trt usubjid;
    where RANDFL eq 'Y' and DSPHASEN=26 and EOTDCDT ne . and (EOSDCDT eq . or
        EOSDCDT>M1P2CUT>.) and dsdecodn not in (. 2) and (VAX101DT ne . or VAX102DT
        ne .);
run;

proc freq data=_temp99;
    format DSDECODN;
    tables _datasrt*_blcksrt*_cat * DSDECODN * _trt / sparse norow nocol nopercnt
        out=_pct9(drop=percent);
run;

proc sort data=_temp9 out=_analcnt9 nodupkey;
    by _datasrt _cat _trt USUBJID;
    where RANDFL eq 'Y';
run;

proc freq data=_analcnt9 noprint;
    tables _datasrt*_cat * _trt / sparse noprint out=_denom9(drop=percent);
run;

data _denomf9;
    length _cat $100;
    _datasrt=1;
    set _bydat1(keep=);
    * All treatment groups ;
    _trt1=0;
    _trt2=0;
    * _CAT is the subgroup variable ;
    _cat="&subv1.";
    output;
    _cat="&subv2.";
    output;
run;

proc transpose data=_denom9 out=_denomin9(drop=_name _label_) prefix=_trt;

```

```

    by _datasrt _cat;
    var count;
    id _trt;
run;

proc sort data=_pct9 out=_expv9 (keep=_datasrt _blcksrt DSDECODN) nodupkey;
    by _datasrt _blcksrt DSDECODN;
run;

proc sort data=_expv9;
    by _datasrt _blcksrt DSDECODN;
run;

data _frame9;
    set _expv9;
    by _datasrt _blcksrt DSDECODN;
    length _cat $100;

    if first._blcksrt then
        _catord=0;
        _catord + 1;
        _trt=1;
        _subcat=1;
        _cat="&subv1.";
        output;
        _subcat=2;
        _cat="&subv2.";
        output;
        _trt=2;
        _subcat=1;
        _cat="&subv1.";
        output;
        _subcat=2;
        _cat="&subv2.";
        output;
run;

proc sort data=_frame9;
    by _datasrt _blcksrt _cat DSDECODN _trt;
run;

proc sort data=_pct9;
    by _datasrt _blcksrt _cat DSDECODN _trt;
run;

data _pct9;
    merge _frame9(in=_inframe) _pct9;
    by _datasrt _blcksrt _cat DSDECODN _trt;

    if _inframe;

    if count=. then
        count=0;
run;

```



```

proc sort data=_pct9;
    by _datasrt _blcksrt DSDECODN;
run;

data _miss9(keep=_datasrt _blcksrt DSDECODN totcount);
    set _pct9;
    where DSDECODN=9998;
    retain totcount;
    by _datasrt _blcksrt DSDECODN;

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

    if last.DSDECODN;
run;

data _pct9(drop=totcount);
    merge _pct9 _miss9;
    by _datasrt _blcksrt DSDECODN;

    if totcount=0 then
        delete;
run;

proc sort data=_denomf9;
    by _datasrt _cat;
run;

proc sort data=_denomin9;
    by _datasrt _cat;
run;

data _denomin9;
    merge _denomf9(in=_inframe) _denomin9;
    by _datasrt _cat;

    if _inframe;
    _blcksrt=2;
run;

proc sort data=_pct9;
    by _datasrt _cat;
run;

data _pct9;
    if 0 then
        set _basemplate;
    merge _denomin9(in=_a) _pct9;
    by _datasrt _cat;

    if _a;
    _varname="DSDECODN ";

```

```

    _vrlabel="Reason for discontinuation from vaccination period ";
    _rwlabel=put(DSDECODN, dsdecod.);

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

    if _catord=. then
        _catord=9997;

run;

proc sort data=_pct9;
    by _datasrt _blcksrt _catord DSDECODN _trt _cat;
run;

data _base9;
    length _catlabl $200;
    set _pct9 end=eof;
    by _datasrt _blcksrt _catord DSDECODN _trt _cat;
    retain _rowsrt 4 _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.);

    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;
                    else
                        _cvalue=trim(_cvalue)||_cpct;
                end;
            else
                _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.DSDECODN 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=8;
    _dptindt=4;

if _trt=2 +1 then
    _trt=9999;

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

/* Crit 10 */
data _anal10;
    length _EVENT10 8;
    length _cat $100;
    set _data1;
    where AGEGR4 is not missing;
    where same and _EVENT10 is not missing;
    _blksrt=3;
    _cnt=1;
    _cat=AGEGR4;

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

proc sort data=_anal10;
    by _datasrt _blksrt _EVENT10 _trt _cat;

```

```

run;

data _subgrpvar;
  set _data1 (keep=AGEGR4);
  where ^missing(AGEGR4);
  format AGEGR4;
run;

proc sql noprint;
  select distinct AGEGR4 into :subv1 - : subv2 from _subgrpvar where AGEGR4 is
    not missing;
quit;

data _temp10;
  set _anal10;
  output;
run;

proc sort data=_temp10 out=_temp910 nodupkey;
  by _datasrt _blcksrt _cat _EVENT10 _trt usubjid;
  ;
run;

proc freq data=_temp910;
  format _EVENT10;
  tables _datasrt*_blcksrt*_cat * _EVENT10 * _trt / sparse norow nocol nopercnt
    out=_pct10(drop=percent);
run;

proc sort data=_temp10 out=_analcnt10 nodupkey;
  by _datasrt _cat _trt USUBJID;
  where RANDFL eq 'Y';
run;

proc freq data=_analcnt10 noprint;
  tables _datasrt*_cat * _trt / sparse noprint out=_denom10(drop=percent);
run;

data _denomf10;
  length _cat $100;
  _datasrt=1;
  set _bydat1(keep=);
  * All treatment groups ;
  _trt1=0;
  _trt2=0;
  * _CAT is the subgroup variable ;
  _cat="&subv1.";
  output;
  _cat="&subv2.";
  output;
run;

proc transpose data=_denom10 out=_denomin10(drop=_name__label_) prefix=_trt;
  by _datasrt _cat;

```

```

    var count;
    id _trt;
run;

data _frame10;
    _datasrt=1;
    set _bydat1(keep=);
    _blcksrt=3;
    length _EVENT10 8;
    length _cat $100;
    _catLabl=" ";
    _trt=1;
    _EVENT10=1;
    _catord=1;
    _subcat=1;
    _cat="&subv1.";
output;
    _subcat=2;
    _cat="&subv2.";
output;
    _trt=2;
    _EVENT10=1;
    _catord=1;
    _subcat=1;
    _cat="&subv1.";
output;
    _subcat=2;
    _cat="&subv2.";
output;
run;

proc sort data=_frame10;
    by _datasrt _blcksrt _cat _EVENT10 _trt;
run;

proc sort data=_pct10;
    by _datasrt _blcksrt _cat _EVENT10 _trt;
run;

data _pct10;
    merge _frame10(in=_inframe) _pct10;
    by _datasrt _blcksrt _cat _EVENT10 _trt;

    if _inframe;

    if count=. then
        count=0;
run;

proc sort data=_pct10;
    by _datasrt _blcksrt _EVENT10;
run;

data _miss10(keep=_datasrt _blcksrt _EVENT10 totcount);

```

```

set _pct10;
where _EVENT10=9998;
retain totcount;
by _datasrt _blcksrt _EVENT10;

if first._EVENT10 then
    totcount=0;
totcount=totcount+count;

if last._EVENT10;

run;

data _pct10(drop=totcount);
merge _pct10 _miss10;
by _datasrt _blcksrt _EVENT10;

if totcount=0 then
    delete;

run;

proc sort data=_denomf10;
by _datasrt _cat;
run;

proc sort data=_denomin10;
by _datasrt _cat;
run;

data _denomin10;
merge _denomf10(in=_inframe) _denomin10;
by _datasrt _cat;

if _inframe;
    _blcksrt=3;

run;

proc sort data=_pct10;
by _datasrt _cat;
run;

data _pct10;
if 0 then
    set _basetemplate;
merge _denomin10(in=_a) _pct10;
by _datasrt _cat;

if _a;
    _varname="_EVENT10 ";
    _vrlabel=" ";
    _rwlabel="Withdrawn from the study before 1-month post(*ESC*){unicode 2013}Dose 2 visit ";

if _EVENT10=9998 then
    do;
        _rwlabel="Missing ";

```

```

        _catord=9998;
    end;
else if _EVENT10=9999 then
    do;
        _rwlabel="Total ";
        _catord=9999;
    end;

    if _catord=. then
        _catord=9997;
run;

proc sort data=_pct10;
    by _datasrt _blcksrt _catord _EVENT10 _trt _cat;
run;

data _base10;
    length _catlabl $200;
    set _pct10 end=eof;
    by _datasrt _blcksrt _catord _EVENT10 _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.);

    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;

            end;

        do;
            if length(_cvalue) < 13 then
                do;
                    substr(_cvalue, 13, 1)='A0'x;

```

```

end;

if first._EVENT10 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=' ';
    _dptindt=0;

  if _trt=2 +1 then
    _trt=9999;

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

```

```

/* Crit 11 */
data _anal11;
  length _EVENT11 8;
  length _cat $100;
  set _data1;
  where AGEGR4 is not missing;
  where same and _EVENT11 is not missing;
  _blcksrt=3;
  _cnt=1;
  _cat=AGEGR4;

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

```

```

proc sort data=_anal11;
  by _datasrt _blcksrt _EVENT11 _trt _cat;
run;

```

```

data _subgrpvar;
  set _data1 (keep=AGEGR4);
  where ^missing(AGEGR4);
  format AGEGR4;
run;

```

```

proc sql noprint;
  select distinct AGEGR4 into :subv1 - : subv2 from _subgrpvar where AGEGR4 is

```

FDA-CBER-2022-5812-0735919


```

        not missing;
quit;

data _temp11;
    set _anal11;
    output;
run;

proc sort data=_temp11 out=_temp911 nodupkey;
    by _datasrt _blcksrt _cat _EVENT11 _trt usubjid;
    ;
run;

proc freq data=_temp911;
    format _EVENT11;
    tables _datasrt*_blcksrt*_cat * _EVENT11 * _trt / sparse norow nocol nopercnt
        out=_pct11(drop=percent);
run;

proc sort data=_temp11 out=_analcnt11 nodupkey;
    by _datasrt _cat _trt USUBJID;
    where RANDFL eq 'Y';
run;

proc freq data=_analcnt11 noprint;
    tables _datasrt*_cat * _trt / sparse noprint out=_denom11(drop=percent);
run;

data _denomf11;
    length _cat $100;
    _datasrt=1;
    set _bydat1(keep=);
    * All treatment groups ;
    _trt1=0;
    _trt2=0;
    * _CAT is the subgroup variable ;
    _cat="&subv1.";
    output;
    _cat="&subv2.";
    output;
run;

proc transpose data=_denom11 out=_denomin11(drop=_name _label_) prefix=_trt;
    by _datasrt _cat;
    var count;
    id _trt;
run;

data _frame11;
    _datasrt=1;
    set _bydat1(keep=);
    _blcksrt=3;
    length _EVENT11 8;
    length _cat $100;

```

```

    _catLabl=" ";
    _trt=1;
    _EVENT11=1;
    _catord=1;
    _subcat=1;
    _cat="&subv1.";
    output;
    _subcat=2;
    _cat="&subv2.";
    output;
    _trt=2;
    _EVENT11=1;
    _catord=1;
    _subcat=1;
    _cat="&subv1.";
    output;
    _subcat=2;
    _cat="&subv2.";
    output;
run;

proc sort data=_frame11;
    by _datasrt _blcksrt _cat _EVENT11 _trt;
run;

proc sort data=_pct11;
    by _datasrt _blcksrt _cat _EVENT11 _trt;
run;

data _pct11;
    merge _frame11(in=_inframe) _pct11;
    by _datasrt _blcksrt _cat _EVENT11 _trt;

    if _inframe;

    if count=. then
        count=0;
run;

proc sort data=_pct11;
    by _datasrt _blcksrt _EVENT11;
run;

data _miss11(keep=_datasrt _blcksrt _EVENT11 totcount);
    set _pct11;
    where _EVENT11=9998;
    retain totcount;
    by _datasrt _blcksrt _EVENT11;

    if first._EVENT11 then
        totcount=0;
    totcount=totcount+count;

    if last._EVENT11;

```

```

run;

data _pct11(drop=totcount);
  merge _pct11 _miss11;
  by _datasrt _blcksrt _EVENT11;

  if totcount=0 then
    delete;
run;

proc sort data=_denomf11;
  by _datasrt _cat;
run;

proc sort data=_denomin11;
  by _datasrt _cat;
run;

data _denomin11;
  merge _denomf11(in=_inframe) _denomin11;
  by _datasrt _cat;

  if _inframe;
  _blcksrt=3;
run;

proc sort data=_pct11;
  by _datasrt _cat;
run;

data _pct11;
  if 0 then
    set _basemplate;
  merge _denomin11(in=_a) _pct11;
  by _datasrt _cat;

  if _a;
  _varname="_EVENT11 ";
  _vrlabel=" ";
  _rwlabel="Withdrawn after Dose 1 and before Dose 2 ";

  if _EVENT11=9998 then
    do;
      _rwlabel="Missing ";
      _catord=9998;
    end;
  else if _EVENT11=9999 then
    do;
      _rwlabel="Total ";
      _catord=9999;
    end;

  if _catord=. then
    _catord=9997;

```

```

run;

proc sort data=_pct11;
    by _datasrt _blcksrt _catord _EVENT11 _trt _cat;
run;

data _base11;
    length _catlabl $200;
    set _pct11 end=eof;
    by _datasrt _blcksrt _catord _EVENT11 _trt _cat;
    retain _rowsrt 1 _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.);

    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;

        do;

            if length(_cvalue) < 13 then
                do;
                    substr(_cvalue, 13, 1)='A0'x;
                end;

            if first._EVENT11 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=4;
    _dptindt=0;

    if _trt=2 +1 then
        _trt=9999;

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

/* Crit 12 */
data _anal12;
    length _EVENT12 8;
    length _cat $100;
    set _data1;
    where AGEGR4 is not missing;
    where same and _EVENT12 is not missing;
    _blcksrt=3;
    _cnt=1;
    _cat=AGEGR4;

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

proc sort data=_anal12;
    by _datasrt _blcksrt _EVENT12 _trt _cat;
run;

data _subgrpvar;
    set _data1 (keep=AGEGR4);
    where ^missing(AGEGR4);
    format AGEGR4;
run;

proc sql noprint;
    select distinct AGEGR4 into :subv1 - : subv2 from _subgrpvar where AGEGR4 is
        not missing;
quit;

data _temp12;
    set _anal12;
    output;
run;

proc sort data=_temp12 out=_temp912 nodupkey;

```

```

    by _datasrt _blcksrt _cat _EVENT12 _trt usubjid;
    ;
run;

proc freq data=_temp912;
    format _EVENT12;
    tables _datasrt*_blcksrt*_cat * _EVENT12 * _trt / sparse norow nocol nopercnt
        out=_pct12(drop=percent);
run;

proc sort data=_temp12 out=_analcnt12 nodupkey;
    by _datasrt _cat _trt USUBJID;
    where RANDFL eq 'Y';
run;

proc freq data=_analcnt12 noprint;
    tables _datasrt*_cat * _trt / sparse noprint out=_denom12(drop=percent);
run;

data _denomf12;
    length _cat $100;
    _datasrt=1;
    set _bydat1(keep=);
    * All treatment groups ;
    _trt1=0;
    _trt2=0;
    * _CAT is the subgroup variable ;
    _cat="&subv1.";
    output;
    _cat="&subv2.";
    output;
run;

proc transpose data=_denom12 out=_denomin12(drop=_name__label_) prefix=_trt;
    by _datasrt _cat;
    var count;
    id _trt;
run;

data _frame12;
    _datasrt=1;
    set _bydat1(keep=);
    _blcksrt=3;
    length _EVENT12 8;
    length _cat $100;
    _catLabl=" ";
    _trt=1;
    _EVENT12=1;
    _catord=1;
    _subcat=1;
    _cat="&subv1.";
    output;
    _subcat=2;
    _cat="&subv2.";

```

```

output;
  _trt=2;
  _EVENT12=1;
  _catord=1;
  _subcat=1;
  _cat="&subv1.";
output;
  _subcat=2;
  _cat="&subv2.";
output;
run;

proc sort data=_frame12;
  by _datasrt _blcksrt _cat _EVENT12 _trt;
run;

proc sort data=_pct12;
  by _datasrt _blcksrt _cat _EVENT12 _trt;
run;

data _pct12;
  merge _frame12(in=_inframe) _pct12;
  by _datasrt _blcksrt _cat _EVENT12 _trt;

  if _inframe;

  if count=. then
    count=0;
run;

proc sort data=_pct12;
  by _datasrt _blcksrt _EVENT12;
run;

data _miss12(keep=_datasrt _blcksrt _EVENT12 totcount);
  set _pct12;
  where _EVENT12=9998;
  retain totcount;
  by _datasrt _blcksrt _EVENT12;

  if first._EVENT12 then
    totcount=0;
  totcount=totcount+count;

  if last._EVENT12;
run;

data _pct12(drop=totcount);
  merge _pct12 _miss12;
  by _datasrt _blcksrt _EVENT12;

  if totcount=0 then
    delete;
run;

```

```

proc sort data=_denomf12;
    by _datasrt _cat;
run;

proc sort data=_denomin12;
    by _datasrt _cat;
run;

data _denomin12;
    merge _denomf12(in=_inframe) _denomin12;
    by _datasrt _cat;

    if _inframe;
    _blcksrt=3;
run;

proc sort data=_pct12;
    by _datasrt _cat;
run;

data _pct12;
    if 0 then
        set _basetemplate;
    merge _denomin12(in=_a) _pct12;
    by _datasrt _cat;

    if _a;
    _varname="_EVENT12 ";
    _vrlabel=" ";
    _rwlabel="Withdrawn after Dose 2 and before 1-month post(*ESC*){unicode 2013}Dose 2 visit ";

    if _EVENT12=9998 then
        do;
            _rwlabel="Missing ";
            _catord=9998;
        end;
    else if _EVENT12=9999 then
        do;
            _rwlabel="Total ";
            _catord=9999;
        end;

    if _catord=. then
        _catord=9997;
run;

proc sort data=_pct12;
    by _datasrt _blcksrt _catord _EVENT12 _trt _cat;
run;

data _base12;
    length _catlabl $200;
    set _pct12 end=eof;

```



```

by _datasrt _blcksrt _cator _EVENT12 _trt _cat;
retain _rowsrt 2 _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.);

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;
        substr(_cvalue, 13, 1)='A0'x;
    end;

if first._EVENT12 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=4;
_dptindt=0;

if _trt=2+1 then

```

```

        _trt=9999;

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

/* Crit 13 */
data _anal13;
    length DSDECODN 8;
    length _cat $100;
    set _data1;
    where AGEGR4 is not missing;
    where same and DSDECODN is not missing;
    _blksrt=3;
    _cnt=1;
    _cat=AGEGR4;

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

proc sort data=_anal13;
    by _datasrt _blksrt DSDECODN _trt _cat;
run;

data _subgrpvar;
    set _data1 (keep=AGEGR4);
    where ^missing(AGEGR4);
    format AGEGR4;
run;

data _temp13;
    set _anal13;
    output;
run;

proc sort data=_temp13 out=_temp913 nodupkey;
    by _datasrt _blksrt _cat DSDECODN _trt usubjid;
    where RANDFL eq 'Y' and DSPHASEN=31 and EOSDCDT ne . and dsdecodn not in (. 2)
        and (VAX101DT ne . or VAX102DT ne .) and COMPLTDT ne EOSDCDT;
run;

proc freq data=_temp913;
    format DSDECODN;
    tables _datasrt*_blksrt*_cat * DSDECODN * _trt / sparse norow nocol nopercnt
        out=_pct13(drop=percent);
run;

proc sort data=_temp13 out=_analcnt13 nodupkey;
    by _datasrt _cat _trt USUBJID;
    where RANDFL eq 'Y';

```

```

run;

proc freq data=_analcnt13 noprint;
    tables _datasrt*_cat * _trt / sparse noprint out=_denom13(drop=percent);
run;

data _denomf13;
    length _cat $100;
    _datasrt=1;
    set _bydat1(keep=);
    * All treatment groups ;
    _trt1=0;
    _trt2=0;
    * _CAT is the subgroup variable ;
    _cat="&subv1.";
    output;
    _cat="&subv2.";
    output;
run;

proc transpose data=_denom13 out=_denomin13(drop=_name__label_) prefix=_trt;
    by _datasrt _cat;
    var count;
    id _trt;
run;

proc sort data=_pct13 out=_expv13 (keep=_datasrt _blcksrt DSDECODN) nodupkey;
    by _datasrt _blcksrt DSDECODN;
run;

proc sort data=_expv13;
    by _datasrt _blcksrt DSDECODN;
run;

data _frame13;
    set _expv13;
    by _datasrt _blcksrt DSDECODN;
    length _cat $100;

    if first._blcksrt then
        _catord=0;
    _catord + 1;
    _trt=1;
    _subcat=1;
    _cat="&subv1.";
    output;
    _subcat=2;
    _cat="&subv2.";
    output;
    _trt=2;
    _subcat=1;
    _cat="&subv1.";
    output;
    _subcat=2;

```

```

    _cat="&subv2.";
    output;
run;

proc sort data=_frame13;
    by _datasrt _blcksrt _cat DSDECODN _trt;
run;

proc sort data=_pct13;
    by _datasrt _blcksrt _cat DSDECODN _trt;
run;

data _pct13;
    merge _frame13(in=_inframe) _pct13;
    by _datasrt _blcksrt _cat DSDECODN _trt;

    if _inframe;

    if count=. then
        count=0;
run;

proc sort data=_pct13;
    by _datasrt _blcksrt DSDECODN;
run;

data _miss13(keep=_datasrt _blcksrt DSDECODN totcount);
    set _pct13;
    where DSDECODN=9998;
    retain totcount;
    by _datasrt _blcksrt DSDECODN;

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

    if last.DSDECODN;
run;

data _pct13(drop=totcount);
    merge _pct13 _miss13;
    by _datasrt _blcksrt DSDECODN;

    if totcount=0 then
        delete;
run;

proc sort data=_denomf13;
    by _datasrt _cat;
run;

proc sort data=_denomin13;
    by _datasrt _cat;
run;

```

```

data _denomin13;
  merge _denomf13(in=_inframe) _denomin13;
  by _datasrt _cat;

  if _inframe;
  _blcksrt=3;
run;

proc sort data=_pct13;
  by _datasrt _cat;
run;

data _pct13;
  if 0 then
    set _basetemplate;
  merge _denomin13(in=_a) _pct13;
  by _datasrt _cat;

  if _a;
  _varname="DSDECODN ";
  _vrlabel="Reason for withdrawal from the study ";
  _rwlabel=put(DSDECODN, dsdecod.);

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

  if _catord=. then
    _catord=9997;
run;

proc sort data=_pct13;
  by _datasrt _blcksrt _catord DSDECODN _trt _cat;
run;

data _base13;
  length _catlabl $200;
  set _pct13 end=eof;
  by _datasrt _blcksrt _catord DSDECODN _trt _cat;
  retain _rowsrt 3 _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.);

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.DSDECODN 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=8;
_dptindt=4;

if _trt=2 +1 then
  _trt=9999;

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

```

```

    _p=2;
run;

/* Set together */
data _final;
    set _base1 _base2 _base3 _base4 _base5 _base6 _base7 _base8 _base9 _base10
        _base11 _base12 _base13;
run;

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

* Collect Sub-group data ;

proc sort data=_final;
    by _trt _subcat;
run;

data _final;
    merge _subGrpData _final(in=_b drop=_colabel);
    by _trt _subcat;

    if _b;
run;

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

data _final;
    set _final;
    drop __trt;

    if _trt=9999 then
        __trt=2 + 1;
    else
        __trt=_trt;

    if __trt=. then
        __trt=1;

    if _subcat in (., 9990, 9999) then
        _subcat=2;

    if _subcat < 9990 then
        _column=_subcat + (__trt - 1) * 2;
    else
        _column=_subcat;
    _colabel=translate(trim(_colabel), '^', ' ');

    if _column=9999 then
        _column=2 + 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;
            _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 '^' & _datatyp='data' then
                _vrlbttag=1;
        end;

    if first._rowsrt then
        do;
            _token=repeat(' ', 99);
            _count=1;
            _token=scan(_rwlablel, _count, "|");

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

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

```



```

;

if _rwlabel ne '' then
    _rwlbttag=1;
    _totline + _rowjump - 1;
end;
_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('_rwlbttag', put(_rwlbttag, 1.));
        call symput('_vrlbttag', put(_vrlbttag, 1.));
    end;

run;

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

data _sph (keep=name _s_col _e_col _splabl);
    length _splabl $ 200 _s_col $ 40 _e_col $ 40 name $ 40;
    _s_col=' ';
    _e_col=' ';
    _splabl=' ';
    name=' ';
    _s_col="TRT1";

```

```

    _e_col="TRT2";
    name=_s_col;
    _splabl="Vaccine Group (as Randomized)~{line}";
    output _sph;
    name=_e_col;
    output _sph;
    _s_col="TRT4";
    _e_col="TRT5";
    name=_s_col;
    _splabl="Vaccine Group (as Randomized)~{line}";
    output _sph;
    name=_e_col;
    output _sph;
run;

data _sph;
    set _sph;
    _s_col_num=input(translate(_s_col, " ", "TRT"), best.);
    _e_col_num=input(translate(_e_col, " ", "TRT"), best.);
run;

proc sort data=_sph (where=( _s_col=name)) out=_span_start;
    by _s_col_num descending _e_col_num;
run;

data _span_start;
    retain _span_hdr_order 1;
    set _span_start;
    by _s_col_num descending _e_col_num;

    if (first._s_col_num) then
        _span_hdr_order=1;
    else
        _span_hdr_order=_span_hdr_order + 1;
run;

proc sort data=_sph (where=( _e_col=name)) out=_span_end;
    by _e_col_num descending _s_col_num;
run;

data _span_end;
    retain _span_hdr_order 1;
    set _span_end;
    by _e_col_num descending _s_col_num;

    if (first._e_col_num) then
        _span_hdr_order=1;
    else
        _span_hdr_order=_span_hdr_order + 1;
run;

data _sph;
    set _span_start _span_end;
run;

```

```

proc sort data=_sph out=_sph nodupkey;
  by _s_col_num descending _e_col_num _s_col _e_col _splabl name;
run;

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

data outdata1;
  length _rvalue $800;
  set _final rson end=eof;
  _rwindt=sum(_indent, _dptindt);

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

  if _trt=. then
    _trt=1;
run;

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

proc sort data=outdata1 out=temp(keep=_column _colabel) nodupkey;
  by _colabel _column;
run;

data temp;
  length _newvar $80;
  set temp;
  by _colabel _column;

  if first._colabel then
    _d=-1;
  _d+1;

  if (first._colabel and last._colabel) then
    _newvar=trim(left(_colabel));
  else
    _newvar=trim(left(_colabel))||repeat(byte(160), _d);
run;

proc sql noprint;
  create table __temp1(drop=_colabel rename=( _newvar=_colabel) drop=_subcat
    rename=( _column=_subcat)) as select a.*, b._newvar from outdata1 as a left
  join temp as b on trim(left(a._colabel))=trim(left(b._colabel)) and

```

```

        a._column=b._column;
quit;

data outdata1;
    set __temp1;
run;

proc sql;
    create table subcat as select distinct 'SUBCAT' as FMTNAME length=8 , _subcat
        as start, tranwrd(_colabel, '^', ' ') as label from outdata1 order by
        fmtname, start;
quit;

data subcat;
    set subcat;
    by fmtname start;

    if start not in (888888) then
        label=trim(label) || "|n~{super b}    (%>";
run;

proc format cntlin=subcat;
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);
        output;
    end;
run;

data outdata1;
    set outdata1(rename=( _cvalue=_cvalue11));
    _fixvar=1;
    _fix2var=1;

    if index(_cvalue11, "(") then
        do;

            if substr(_cvalue11, length(_cvalue11), 1) ne ")" then
                _cvalue=_cvalue11||"";
            else
                _cvalue=_cvalue11;
        end;
    else if not missing(_cvalue11) then
        _cvalue=_cvalue11;
run;

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

```

```

run;

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

data _pre_transposed;
    set _pre_transposed;

    if _trt=9999 then
        _trt=2 +1;
run;

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

proc transpose data=_pre_transposed out=_cntsort (drop=_name_) prefix=cntsort;
    by _fixvar _fix2var _datasrt _blcksrt _rowsrt _rvalue;
    var count;
run;

data _column_transposed;
    merge _column_transposed(in=a) _cntsort;
    by _fixvar _fix2var _datasrt _blcksrt _rowsrt _rvalue;

    if a;
    cntsort=0;
    cntsort=cntsort+input(cntsort1, best.);
    drop cntsort1;
    cntsort=cntsort+input(cntsort2, best.);
    drop cntsort2;
    cntsort=cntsort+input(cntsort3, best.);
    drop cntsort3;
    cntsort=cntsort+input(cntsort4, best.);
    drop cntsort4;

proc sort;
    by _fixvar _fix2var _datasrt _blcksrt descending cntsort _rvalue _rowsrt;
run;

data _column_transposed;
    set _column_transposed;
    by _fixvar _fix2var _datasrt _blcksrt descending cntsort _rvalue _rowsrt;

    if _blcksrt=2 then
        do;

            if _rowsrt>4.5 then
                do;

```

```

        if first._blcksrt then
            srt=1;
        else if _rvalue ne lag(_rvalue) then
            srt+1;
        _rowsrt=4.5+srt;

        if index(upcase(_rvalue), 'OTHER')>0 then
            _rowsrt=999;

        if cntsort=0 then
            delete;
    end;
end;

if _blcksrt=3 then
    do;

        if _rowsrt>3.5 then
            do;

                if first._blcksrt then
                    srt=1;
                else if _rvalue ne lag(_rvalue) then
                    srt+1;
                _rowsrt=3.5+srt;

                if index(upcase(_rvalue), 'OTHER')>0 then
                    _rowsrt=999;

                if cntsort=0 then
                    delete;
            end;
        end;
    drop srt cntsort;

proc sort;
    by _fixvar _fix2var _datasrt _blcksrt _rowsrt _rvalue;
run;

proc contents data=_column_transposed
    out=_col_labels (where=(upcase(name)= "TRT") keep=name) noprint;
run;

data _col_labels;
    length name $ 40;
    set _col_labels end=eof;
    _sort_order=_n_;
    name=upcase(name);

    if eof then
        call symput("_max_trt", left(put(_sort_order, best.)));
run;

proc sort data=_col_labels out=_col_labels;

```

```

    by name;
run;

proc sort data=_sph out=_sph;
    by name;
run;

data _final_sph;
    merge _col_labels (in=a) _sph;
    by name;

    if a;
run;

proc sort data=_final_sph out=_final_sph nodup;
    by _sort_order _span_hdr_order;
run;

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

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

/* Output report */
ods escapechar="~";
ods html file="&outtable.";
title1 j=1 "Disposition of All Randomized Subjects Through 1 Month After Dose 2 (*ESC*){unicode 2013}";
title2 "Subjects 12 Through 15 and 16 Through 25 Years of Age";
footnote1 "Note: Human immunodeficiency virus (HIV)-positive subjects are included in this summary but not included in the analyses of the overall study objectives. ";
footnote2 "Note: Subjects randomized but did not sign informed consent or had a significant quality event due to lack of PI oversight are not included in any analysis population.";
footnote3 "a.(*ESC*){nbspspace 5}N = number of randomized subjects in the specified group. This value is the denominator for the percentage calculations. ";
footnote4 "b.(*ESC*){nbspspace 5}n = Number of subjects with the specified characteristic. ";

proc report data=report nowd list missing contents="" split="" nocompletecols
    nocompleterows style(report)={} style(header)={} style(column)={};
    column _fixvar _fix2var _datasrt _blcksrt _rowsrt (" " " " "" _rvalue)
        (("Vaccine Group (as Randomized)~{line}" TRT1, _subcat TRT2, _subcat) _dummy);
    define _fixvar / group noprint;
    define _fix2var / group noprint;
    define _subcat / across order=internal ''
        format=subcat. style(header)={just=center};
    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=25mm
        rightmargin=18px} style(header)={just=left} left;
;

```

```

define _dummy / sum noprint;
define TRT1 / group nozero "BNT162b2 (30 (*ESC*){unicode 03BC}g)~{line}"
    spacing=2 style(header)={just=center} center;
define TRT2 / group nozero "Placebo~{line}" spacing=2
    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 markup close;
ods HTML close;

proc printto;
run;

```