

```

*****
** Program Name   : adsl_s005_demo_ped_saf.sas
** Date Created  : 10Mar2021
** Programmer Name : (b) (6)
** Purpose       : Create adsl_s005_demo_ped_saf
** Input data    : adsl
** Output file   : adsl_s005_demo_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_s005_demo_ped_saf.log;
%let outtable=&outpath./output/adsl_s005_demo_ped_saf.html;
*****
* Clean *;
*****

options mprint mlogic symbolgen;
title;
footnote;

proc delete data=work._all_;
run;

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

proc format;
  value cov 1="Positive" 2="Negative";
  value sars 1="Positive(*ESC*){super c}" 2="Negative(*ESC*){super d}";
  value cd 1="<200 cells/mm(*ESC*){super 3}"
    2="200-500 cells/mm(*ESC*){super 3}" 3=">500 cells/mm(*ESC*){super 3}";
  value rna 1="<50 copies/mL" 2="(*ESC*){unicode 2265}50 copies/mL";
  value sex 1='Male' 2='Female';
  value arace 1='White' 2='Black or African American'
    3='American Indian or Alaska Native' 4='Asian'
    5='Native Hawaiian or other Pacific Islander' 6='Multiracial'
    7='Not reported' 8='Unknown' 999='All others~{super c}';
  value ethnic 1='Hispanic/Latino' 2='Non-Hispanic/non-Latino' 3='Not reported'
    4='Unknown';
  value RANDAGE 1='12-15 Years' 2='16-55 Years' 3='18-55 Years' 4='65-85 Years'
    5='>55 Years';
  value Raciald 1="Indian Subcontinent Asian" 10="African Caribbean"
    11="Saudi Arabian" 12="Malay" 13="Filipino" 14="Vietnamese"
    15="Australian Aboriginal" 16="Torres Strait Islander" 17="Han Chinese"
    18="Non-Han Chinese" 19="Ashkenazi Jew" 2="Southeast Asian"
    3="Far East Asian" 4="Japanese American" 5="Japanese" 6="Korean" 7="Chinese"

```

```

      8="African" 9="African American" 999="Other";
value BMICAT 1="Underweight ((*ESC*){Unicode 003C} 18.5 kg/m~{super 2})" 2=" Normal weight ((*ESC*)
{Unicode 2265} 18.5 kg/m~{super 2} - 24.9 kg/m~{super 2})"
      3="Overweight ((*ESC*){Unicode 2265} 25.0 kg/m~{super 2} - 29.9 kg/m~{super 2})"
      4="Obese ((*ESC*){Unicode 2265} 30.0 kg/m~{super 2})" 5="Missing";

run;

data adsl;
  set DATVPROT.ADSL(rename=(ethnic=ethnic1));
  length ethnic $50;

  if covblst="POS" then
    do;
      covblst="Positive";
      covblstc="Positive(*ESC*){super c}";
      covblstn=1;
    end;
  else if covblst="NEG" then
    do;
      covblst="Negative";
      covblstc="Negative(*ESC*){super d}";
      covblstn=2;
    end;
  else
    covblstn=.;

  if upcase(ethnic1)='NOT HISPANIC OR LATINO' then
    ethnic='Non-Hispanic/Non-Latino';
  else if upcase(ethnic1)='HISPANIC OR LATINO' then
    ethnic='Hispanic/Latino';
  else if upcase(ethnic1)='NOT REPORTED' then
    ethnic='Not Reported';

run;

data adsl;
  set adsl;
  length countryx $50;

  if country='ARG' then
    countryx='Argentina';
  else if country='BRA' then
    countryx='Brazil';
  else if country='DEU' then
    countryx='Germany';
  else if country='TUR' then
    countryx='Turkey';
  else if country='USA' then
    countryx='USA';
  else if country='ZAF' then
    countryx='South Africa';
  else
    countryx='Others';

run;

```

```

data adsl;
  set adsl;

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

  if racialdn=999 then
    racialdn=.;
run;

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

data __trtmap;
  length trtcode trtdec $100;

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

  if vtype(TRTARN)='C' then
    trtcode=tranwrd(compbl(quote("1")), ' ', "" );
  else
    trtcode="1";
  trtdec="12-15 Years";
  trtvar="TRTARN";
  trtlbl="TRTAR";
  output;
  trtval=2;

  if vtype(TRTARN)='C' then
    trtcode=tranwrd(compbl(quote("2")), ' ', "" );
  else
    trtcode="2";
  trtdec="16-25 Years";
  trtvar="TRTARN";
  trtlbl="TRTAR";
  output;
  trtval=3;

  if vtype(TRTARN)='C' then
    trtcode=tranwrd(compbl(quote("3")), ' ', "" );
  else
    trtcode="3";
  trtdec="12-15 Years";

```

```

trtvar="TRTARN";
trtlbl="TRTAR";
output;
trtval=4;

if vtype(TRTARN)='C' then
    trtcode=tranwrd(compbl(quote("4")), ' ', "" );
else
    trtcode="4";
trtdecdec="16-25 Years";
trtvar="TRTARN";
trtlbl="TRTAR";
output;
stop;
run;

data _null_;
length cc $8 path $256;
rc=filename(cc, 'TEMP');
path=pathname(cc);
rc=filename(cc);
call symputX('INCPATH1', quote(strip(path)));
file dummy filevar=path;

do _n_=1 by 1 until(eof);
    set __trtmap(in=in1) end=eof;

    if in1 then
        do;
            put +6 'if ' trtvar 'in (' trtcode +(-1) ')' @;
            put +1 'then do; ' 'newtrtn =' trtval +(-1) ';' @;
            put +1 'newtrtn = coalescec("'" trtdecdec +(-1) ""',' trtlbl +(-1)
                '); output; end;';
        end;
    end;
stop;
run;

data g_adsl_dsin;
set g_adsl_dsin;

if TRTARN in (1) then
    do;
        newtrtn=1;
        newtrtn=coalescec("12-15 Years", TRTAR);
        output;
    end;

if TRTARN in (2) then
    do;
        newtrtn=2;
        newtrtn=coalescec("16-25 Years", TRTAR);
        output;
    end;

```

```

if TRTARN in (3) then
  do;
    newtrtn=3;
    newtrt=coalescec("12-15 Years", TRTAR);
    output;
  end;

if TRTARN in (4) then
  do;
    newtrtn=4;
    newtrt=coalescec("16-25 Years", TRTAR);
    output;
  end;

run;

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

data _stdft2(compress=no);
  length model $200 mark $5;
  index=0;
  model=' ';
  mark=' ';
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 g_adsl_dsin;
  where (NEWTRTN is not missing);
run;

proc sql noprint;
  select put(nobs - delobs, 12.) into :_nobs from dictionary.tables
    where (libname="WORK" and memname="_DATA1");
  select setting into :miss from dictionary.options where
    upcase(optname)="MISSING";
quit;

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

```

```

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

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

    if first.NEWTRTN then
        do;

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

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

proc sql noprint;
    select compress(put(count(*), 5.) ) into : _trt1 - : _trt4 from (select distinct
        USUBJID, _trt from _data1 where NEWTRTN is not missing) group by _trt;
    select compress(put(count(*), 5.) ) into : _trt5 from (select distinct USUBJID
        from _data1 where NEWTRTN is not missing);
quit;

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;
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;

data _null_;
  set _data1 end=eof;

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

data _anall;
  length SEXN 8;
  set _data1;

  if SEXN=. then
    SEXN=9998;
  _blcksrt=1;
  _cnt=1;
  _cat=1;

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

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

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

data _temp1;
  set _anall;
  output;
run;

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

proc freq data=_temp91;

```

```

format SEXN;
tables _datasrt*_blcksrt*_cat * SEXN * _trt / sparse norow nocol nopercnt
      out=_pct1(drop=percent);
run;

proc sort data=_anall 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;
  _trt=4;
  count=&_trt4;
  output;
run;

data _denomf1;
  _datasrt=1;
  set _bydat1(keep=);
  * All treatment groups ;
  _trt1=0;
  _trt2=0;
  _trt3=0;
  _trt4=0;
  * _CAT is the subgroup variable ;
  _cat=1;
  output;
run;

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

proc sql noprint;
  select count(distinct SEXN) into : totexpv from _anall;
  select distinct SEXN into :expv1 - :expv2 from _anall order by SEXN;
quit;

data _frame1;
  _datasrt=1;
  set _bydat1(keep=);

```



```

    _blcksrt=1;
length SEXN 8;
    _catLabl=" ";
    _trt=1;
SEXN=1;
    _catord=1;
    _cat=1;
output;
    _trt=2;
SEXN=1;
    _catord=1;
    _cat=1;
output;
    _trt=3;
SEXN=1;
    _catord=1;
    _cat=1;
output;
    _trt=4;
SEXN=1;
    _catord=1;
    _cat=1;
output;
    _catLabl=" ";
    _trt=1;
SEXN=2;
    _catord=2;
    _cat=1;
output;
    _trt=2;
SEXN=2;
    _catord=2;
    _cat=1;
output;
    _trt=3;
SEXN=2;
    _catord=2;
    _cat=1;
output;
    _trt=4;
SEXN=2;
    _catord=2;
    _cat=1;
output;
run;

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

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

```

```

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

  if _inframe;

  if count=. then
    count=0;
run;

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

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

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

  if last.SEXN;
run;

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

  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="SEXN ";
  _vrlabel="Sex ";
  _rwlabel=put(SEXN, sex.);

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

  if _catord=. then
    _catord=9997;
run;

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

data _base1;
  length _catlabl $200;
  set _pct1 end=eof;
  by _datasrt _blcksrt _catord SEXN _trt _cat;
  retain _rowsrt 0 _rowmax 0;
  array _trtcnt(*) _trt1-_trt5;
  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.SEXN 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=3;
_dptindt=0;

if _trt=4 +1 then
    _trt=9999;

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

data _null_;
set _data1 end=eof;

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

data _anal2;
length ARACEN 8;
set _data1;
where same and ARACEN is not missing;
_bklsrt=2;

```

```

    _cnt=1;
    _cat=1;

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

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

data _temp2;
    set _anal2;
    output;
run;

proc sort data=_temp2 out=_temp92 nodupkey;
    by _datasrt _blcksrt _cat ARACEN _trt USUBJID;
run;

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

proc sort data=_anal2 out=_denom2(keep=_datasrt _cat) nodupkey;
    by _datasrt _cat;
run;

data _denom2;
    set _denom2;
    by _datasrt _cat;
    label count='count';
    _trt=1;
    count=&_trt1;
    output;
    _trt=2;
    count=&_trt2;
    output;
    _trt=3;
    count=&_trt3;
    output;
    _trt=4;
    count=&_trt4;
    output;
run;

data _denomf2;
    _datasrt=1;
    set _bydat1(keep=);
    * All treatment groups ;
    _trt1=0;

```

```

    _trt2=0;
    _trt3=0;
    _trt4=0;
    * _CAT is the subgroup variable ;
    _cat=1;
    output;
run;

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

proc sql noprint;
  select count(distinct ARACEN) into : totexpv from _anal2;
  select distinct ARACEN into :expv1 - :expv7 from _anal2 order by ARACEN;
quit;

data _frame2;
  _datasrt=1;
  set _bydat1(keep=);
  _blcksrt=2;
  length ARACEN 8;
  _catLabl=" ";
  _trt=1;
  ARACEN=1;
  _catord=1;
  _cat=1;
  output;
  _trt=2;
  ARACEN=1;
  _catord=1;
  _cat=1;
  output;
  _trt=3;
  ARACEN=1;
  _catord=1;
  _cat=1;
  output;
  _trt=4;
  ARACEN=1;
  _catord=1;
  _cat=1;
  output;
  _catLabl=" ";
  _trt=1;
  ARACEN=2;
  _catord=2;
  _cat=1;
  output;
  _trt=2;
  ARACEN=2;
  _catord=2;

```

```

_cat=1;
output;
_trt=3;
ARACEN=2;
_catord=2;
_cat=1;
output;
_trt=4;
ARACEN=2;
_catord=2;
_cat=1;
output;
_catLabl=" ";
_trt=1;
ARACEN=3;
_catord=3;
_cat=1;
output;
_trt=2;
ARACEN=3;
_catord=3;
_cat=1;
output;
_trt=3;
ARACEN=3;
_catord=3;
_cat=1;
output;
_trt=4;
ARACEN=3;
_catord=3;
_cat=1;
output;
_catLabl=" ";
_trt=1;
ARACEN=4;
_catord=4;
_cat=1;
output;
_trt=2;
ARACEN=4;
_catord=4;
_cat=1;
output;
_trt=3;
ARACEN=4;
_catord=4;
_cat=1;
output;
_trt=4;
ARACEN=4;
_catord=4;
_cat=1;
output;

```

```

_catLabl=" ";
_trt=1;
ARACEN=5;
_catord=5;
_cat=1;
output;
_trt=2;
ARACEN=5;
_catord=5;
_cat=1;
output;
_trt=3;
ARACEN=5;
_catord=5;
_cat=1;
output;
_trt=4;
ARACEN=5;
_catord=5;
_cat=1;
output;
_catLabl=" ";
_trt=1;
ARACEN=6;
_catord=6;
_cat=1;
output;
_trt=2;
ARACEN=6;
_catord=6;
_cat=1;
output;
_trt=3;
ARACEN=6;
_catord=6;
_cat=1;
output;
_trt=4;
ARACEN=6;
_catord=6;
_cat=1;
output;
_catLabl=" ";
_trt=1;
ARACEN=7;
_catord=7;
_cat=1;
output;
_trt=2;
ARACEN=7;
_catord=7;
_cat=1;
output;
_trt=3;

```



```

ARACEN=7;
_catord=7;
_cat=1;
output;
_trt=4;
ARACEN=7;
_catord=7;
_cat=1;
output;
run;

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

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

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

  if _inframe;

  if count=. then
    count=0;
run;

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

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

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

  if last.ARACEN;
run;

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

  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=2;
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="ARACEN ";
    _vrlabel="Race ";
    _rwlabel=put(ARACEN, arace.);

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

    if _catord=. then
        _catord=9997;
run;

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

data _base2;
    length _catlabl $200;
    set _pct2 end=eof;
    by _datasrt _blcksrt _catord ARACEN _trt _cat;

```

```

retain _rowsrt 0 _rowmax 0;
array _trtcnt(*) _trt1-_trt5;
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;

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

if first.ARACEN 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=3;
_dptindt=0;

if _trt=4 +1 then
    _trt=9999;

```

```

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

data _null_;
    set _data1 end=eof;

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

data _anal3;
    length RACIALDN 8;
    set _data1;
    where same and RACIALDN is not missing;
    _blcksrt=3;
    _cnt=1;
    _cat=1;

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

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

data _temp3;
    set _anal3;
    output;
run;

proc sort data=_temp3 out=_temp93 nodupkey;
    by _datasrt _blcksrt _cat RACIALDN _trt USUBJID;
run;

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

proc sort data=_anal3 out=_denom3(keep=_datasrt _cat) nodupkey;
    by _datasrt _cat;
run;

data _denom3;
    set _denom3;
    by _datasrt _cat;
    label count='count';

```

```

    _trt=1;
    count=&_trt1;
    output;
    _trt=2;
    count=&_trt2;
    output;
    _trt=3;
    count=&_trt3;
    output;
    _trt=4;
    count=&_trt4;
    output;
run;

data _denomf3;
    _datasrt=1;
    set _bydat1(keep=);
    * All treatment groups ;
    _trt1=0;
    _trt2=0;
    _trt3=0;
    _trt4=0;
    * _CAT is the subgroup variable ;
    _cat=1;
    output;
run;

proc transpose data=_denom3 out=_denomin3(drop=_name__label_) prefix=_trt;
    by _datasrt _cat;
    var count;
    id _trt;
run;

proc sql noprint;
    select count(distinct RACIALDN) into :totexpv from _anal3;
    select distinct RACIALDN into :expv1 - :expv1 from _anal3 order by RACIALDN;
quit;

data _frame3;
    _datasrt=1;
    set _bydat1(keep=);
    _blcksrt=3;
    length RACIALDN 8;
    _catLabl=" ";
    _trt=1;
    RACIALDN=5;
    _catord=1;
    _cat=1;
    output;
    _trt=2;
    RACIALDN=5;
    _catord=1;
    _cat=1;
    output;

```

```

    _trt=3;
    RACIALDN=5;
    _catord=1;
    _cat=1;
    output;
    _trt=4;
    RACIALDN=5;
    _catord=1;
    _cat=1;
    output;
run;

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

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

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

    if _inframe;

    if count=. then
        count=0;
run;

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

data _miss3(keep=_datasrt _blcksrt RACIALDN totcount);
    set _pct3;
    where RACIALDN=9998;
    retain totcount;
    by _datasrt _blcksrt RACIALDN;

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

    if last.RACIALDN;
run;

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

    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=3;
run;

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

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

    if _a;
    _varname="RACIALDN ";
    _vrlabel="Racial designation ";
    _rwlabel=put(RACIALDN, raciald.);

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

    if _catord=. then
        _catord=9997;
run;

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

data _base3;
    length _catlabl $200;
    set _pct3 end=eof;

```

```

by _datasrt _blcksrt _cator RACIALDN _trt _cat;
retain _rowsrt 0 _rowmax 0;
array _trtcnt(*) _trt1-_trt5;
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.RACIALDN 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=3;

```



```

_dptindt=0;

if _trt=4 +1 then
    _trt=9999;

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

data _null_;
    set _data1 end=eof;

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

data _anal4;
    length ETHNICN 8;
    set _data1;
    where same and ETHNICN is not missing;
    _blcksrt=4;
    _cnt=1;
    _cat=1;

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

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

data _temp4;
    set _anal4;
    output;
run;

proc sort data=_temp4 out=_temp94 nodupkey;
    by _datasrt _blcksrt _cat ETHNICN _trt USUBJID;
run;

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

proc sort data=_anal4 out=_denom4(keep=_datasrt _cat) nodupkey;
    ;
    by _datasrt _cat;
run;

```

```

data _denom4;
  set _denom4;
  by _datasrt _cat;
  label count='count';
  _trt=1;
  count=&_trt1;
  output;
  _trt=2;
  count=&_trt2;
  output;
  _trt=3;
  count=&_trt3;
  output;
  _trt=4;
  count=&_trt4;
  output;
run;

data _denomf4;
  _datasrt=1;
  set _bydat1(keep=);
  * All treatment groups ;
  _trt1=0;
  _trt2=0;
  _trt3=0;
  _trt4=0;
  * _CAT is the subgroup variable ;
  _cat=1;
  output;
run;

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

proc sql noprint;
  select count(distinct ETHNICN) into :totexpv from _anal4;
  select distinct ETHNICN into :expv1 - :expv3 from _anal4 order by ETHNICN;
quit;

data _frame4;
  _datasrt=1;
  set _bydat1(keep=);
  _blcksrt=4;
  length ETHNICN 8;
  _catLabl=" ";
  _trt=1;
  ETHNICN=1;
  _catord=1;
  _cat=1;
  output;

```

```

_trt=2;
ETHNICN=1;
_catord=1;
_cat=1;
output;
_trt=3;
ETHNICN=1;
_catord=1;
_cat=1;
output;
_trt=4;
ETHNICN=1;
_catord=1;
_cat=1;
output;
_catLabl=" ";
_trt=1;
ETHNICN=2;
_catord=2;
_cat=1;
output;
_trt=2;
ETHNICN=2;
_catord=2;
_cat=1;
output;
_trt=3;
ETHNICN=2;
_catord=2;
_cat=1;
output;
_trt=4;
ETHNICN=2;
_catord=2;
_cat=1;
output;
_catLabl=" ";
_trt=1;
ETHNICN=3;
_catord=3;
_cat=1;
output;
_trt=2;
ETHNICN=3;
_catord=3;
_cat=1;
output;
_trt=3;
ETHNICN=3;
_catord=3;
_cat=1;
output;
_trt=4;
ETHNICN=3;

```

```

    _catord=3;
    _cat=1;
    output;
run;

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

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

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

    if _inframe;

    if count=. then
        count=0;
run;

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

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

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

    if last.ETHNICN;
run;

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

    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=4;
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="ETHNICN ";
  _vrlabel="Ethnicity ";
  _rwlabel=put(ETHNICN, ethnic.);

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

  if _catord=. then
    _catord=9997;
run;

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

data _base4;
  length _catlabl $200;
  set _pct4 end=eof;
  by _datasrt _blcksrt _catord ETHNICN _trt _cat;
  retain _rowsrt 0 _rowmax 0;
  array _trcnt(*) _trt1-_trt5;
  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;

  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.ETHNICN 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=3;
_dptindt=0;

if _trt=4 +1 then
  _trt=9999;

```

```

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

data _null_;
    set _data1 end=eof;

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

data _anal5;
    length COUNTRYX $50;
    set _data1;
    where same and COUNTRYX is not missing;
    _blksrt=5;
    _cnt=1;
    _cat=1;

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

proc sort data=_anal5;
    by _datasrt _blksrt COUNTRYX _trt _cat;
run;

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

data _temp5;
    set _anal5;
    output;
run;

proc sort data=_temp5 out=_temp95 nodupkey;
    by _datasrt _blksrt _cat COUNTRYX _trt USUBJID;
run;

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

proc sort data=_anal5 out=_denom5(keep=_datasrt _cat) nodupkey;
    by _datasrt _cat;
run;

data _denom5;
    set _denom5;

```

```

by _datasrt _cat;
label count='count';
_trt=1;
count=&_trt1;
output;
_trt=2;
count=&_trt2;
output;
_trt=3;
count=&_trt3;
output;
    _trt=4;
count=&_trt4;
output;
run;

data _denomf5;
    _datasrt=1;
    set _bydat1(keep=);
    * All treatment groups ;
    _trt1=0;
    _trt2=0;
    _trt3=0;
    _trt4=0;
    * _CAT is the subgroup variable ;
    _cat=1;
    output;
run;

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

proc sql noprint;
    select count(distinct COUNTRYX) into :totexpv from _anal5;
    select distinct COUNTRYX into :expv1 - :expv6 from _anal5 order by COUNTRYX;
quit;

data _frame5;
    _datasrt=1;
    set _bydat1(keep=);
    _blcksrt=5;
    length COUNTRYX $50;
    _catLabl=" ";
    _trt=1;
    COUNTRYX="Argentina ";
    _catord=1;
    _cat=1;
    output;
    _trt=2;
    COUNTRYX="Argentina ";
    _catord=1;

```



```

_cat=1;
output;
_trt=3;
COUNTRYX="Argentina ";
_catord=1;
_cat=1;
output;
_trt=4;
COUNTRYX="Argentina ";
_catord=1;
_cat=1;
output;
_catLabl=" ";
_trt=1;
COUNTRYX="Brazil ";
_catord=2;
_cat=1;
output;
_trt=2;
COUNTRYX="Brazil ";
_catord=2;
_cat=1;
output;
_trt=3;
COUNTRYX="Brazil ";
_catord=2;
_cat=1;
output;
_trt=4;
COUNTRYX="Brazil ";
_catord=2;
_cat=1;
output;
_catLabl=" ";
_trt=1;
COUNTRYX="Germany ";
_catord=3;
_cat=1;
output;
_trt=2;
COUNTRYX="Germany ";
_catord=3;
_cat=1;
output;
_trt=3;
COUNTRYX="Germany ";
_catord=3;
_cat=1;
output;
_trt=4;
COUNTRYX="Germany ";
_catord=3;
_cat=1;
output;

```

```

_catLabl=" ";
_trt=1;
COUNTRYX="South Africa ";
_catord=4;
_cat=1;
output;
_trt=2;
COUNTRYX="South Africa ";
_catord=4;
_cat=1;
output;
_trt=3;
COUNTRYX="South Africa ";
_catord=4;
_cat=1;
output;
_trt=4;
COUNTRYX="South Africa ";
_catord=4;
_cat=1;
output;
_catLabl=" ";
_trt=1;
COUNTRYX="Turkey ";
_catord=5;
_cat=1;
output;
_trt=2;
COUNTRYX="Turkey ";
_catord=5;
_cat=1;
output;
_trt=3;
COUNTRYX="Turkey ";
_catord=5;
_cat=1;
output;
_trt=4;
COUNTRYX="Turkey ";
_catord=5;
_cat=1;
output;
_catLabl=" ";
_trt=1;
COUNTRYX="USA ";
_catord=6;
_cat=1;
output;
_trt=2;
COUNTRYX="USA ";
_catord=6;
_cat=1;
output;
_trt=3;

```

```

COUNTRYX="USA ";
_catord=6;
_cat=1;
output;
_trt=4;
COUNTRYX="USA ";
_catord=6;
_cat=1;
output;
run;

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

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

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

  if _inframe;

  if count=. then
    count=0;
run;

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

data _miss5(keep=_datasrt _blcksrt COUNTRYX totcount);
  set _pct5;
  where COUNTRYX='ZZZY';
  retain totcount;
  by _datasrt _blcksrt COUNTRYX;

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

  if last.COUNTRYX;
run;

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

  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=5;
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="COUNTRYX ";
    _vrlabel="Country ";
    _rwlabel=put(COUNTRYX, country.);

    if COUNTRYX='ZZZY' then
        do;
            _rwlabel="Missing ";
            _catord=9998;
        end;
    else if COUNTRYX='ZZZZ' then
        do;
            _rwlabel="Total ";
            _catord=9999;
        end;

    if _catord=. then
        _catord=9997;
run;

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

data _base5;
    length _catlabl $200;
    set _pct5 end=eof;
    by _datasrt _blcksrt _catord COUNTRYX _trt _cat;

```

```

retain _rowsrt 0 _rowmax 0;
array _trtcnt(*) _trt1-_trt5;
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;

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

if first.COUNTRYX 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=3;
_dptindt=0;

if _trt=4 +1 then
    _trt=9999;

```

```

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

data _anal6;
    set _data1;
    where _trt > 0;
    _blcksrt=6;
    output;
run;

proc sort data=_anal6;
    by _datasrt _blcksrt _trt;
run;

proc univariate data=_anal6 noprint;
    by _datasrt _blcksrt _trt;
    var AGETR01;
    output out=_msum6 CSS=CSS CV=CV KURTOSIS=KURTOSIS MAX=MAX MEAN=MEAN N=N
        MIN=MIN MODE=MODE RANGE=RANGE NMISS=NMISS NOBS=NOBS STDMEAN=STDMEAN
        SKEWNESS=SKEWNESS STD=STD USS=USS SUM=SUM VAR=VAR MEDIAN=MEDIAN P1=P1
P5=P5
        P10=P10 P90=P90 P95=P95 P99=P99 Q1=Q1 Q3=Q3 QRANGE=QRANGE GINI=GINI MAD=MAD
        QN=QN SN=SN STD_GINI=STD_GINI STD_MAD=STD_MAD STD_QN=STD_QN
        STD_QRANGE=STD_QRANGE STD_SN=STD_SN NORMAL=NORMAL PROBN=PROBN
MSIGN=MSIGN
        PROBM=PROBM SIGNRANK=SIGNRANK PROBS=PROBS T=T PROBT=PROBT;
run;

data _frame6;
    set _bydat1(keep=);
    _datasrt=1;
    _blcksrt=6;
    _catord=1;
    _trt=1;
    _cat=1;
    output;
    _trt=2;
    _cat=1;
    output;
    _trt=3;
    _cat=1;
    output;
    _trt=4;
    _cat=1;
    output;
run;

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

```

```

data _msum6;
  merge _msum6_frame6;
  by _datasrt _blcksrt _trt;
run;

data _result1_6;
  if 0 then
    set _basetemplate;
  set _msum6 end=eof;
  _rowsrt=0 + 1;
  _rwlabel="Mean (SD) ";
  _cvalue=' ';
  _nvalue=.;

  if mean ne . and std ne . then
    do;
      _cValue=strip(put(mean, 5.1) ) || ' (' || strip(put(std, 5.2) ) || ');'
    end;
  else if mean eq . then
    _cValue="-" || ' (' || "-" || ');'
  else if std eq . then
    do;
      _cValue=strip(put(mean, 5.1) ) || ' (' || "-" || ');'
    end;
  output;
  _rowsrt=0 + 2;
  _rwlabel="Median ";
  _cvalue=' ';
  _nvalue=.;
  _nvalue=MEDIAN;

  if MEDIAN ne . then
    _cValue=strip(put(MEDIAN, 5.1) );
  else
    _cValue="-";
  output;
  _rowsrt=0 + 3;
  _rwlabel="Min, max ";
  _cvalue=' ';
  _nvalue=.;
  _cValue=' ';

  if min ^=. & max ^=. then
    do;
      _cValue=trim(_cvalue) || ' (' || strip(put(min, 5.0)
        )|| ', ' || strip(put(max, 5.0) )||)';
    end;
  else if min= . & max= . then
    do;
      _cValue=trim(_cvalue) || ' (' || "-" || ', ' || "-" ||)';
    end;
  _cValue=compbl(_cValue);
  output;

```

```

run;

data _logresult1_6;
  if 0 then
    set _basetemplate;
  stop;
run;

data _result2_6;
  if 0 then
    set _basetemplate;
  stop;
run;

data _logresult2_6;
  if 0 then
    set _basetemplate;
  stop;
run;

data _base6;
  set _result1_6 _result2_6 _logresult1_6 _logresult2_6;

  if _trt=5 then
    _trt=9999;
  _varname="AGETR01";
  _vrlabel="Age at vaccination (years) ";
  _datatype='data';
  _module='msumstat';
  _indent=5;
  _rowjump=1;
  _dptindt=0;
run;

proc sort data=_base6;
  by _datasrt _blcksrt _rowsrt;
run;

data _null_;
  set _data1 end=eof;

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

data _anal7;
  length COVBLSTN 8;
  set _data1;
  if COVBLSTN = . then COVBLSTN = 9998;
  _blcksrt=7;
  _cnt=1;
  _cat=1;

  if _trt <=0 then

```



```

        delete;
    output;
run;

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

data _temp7;
    set _anal7;
    output;
run;

proc sort data=_temp7 out=_temp97 nodupkey;
    by _datasrt _blcksrt _cat COVBLSTN _trt USUBJID;
run;

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

proc sort data=_anal7 out=_denom7(keep=_datasrt _cat) nodupkey;
    by _datasrt _cat;
run;

data _denom7;
    set _denom7;
    by _datasrt _cat;
    label count='count';
    _trt=1;
    count=&_trt1;
    output;
    _trt=2;
    count=&_trt2;
    output;
    _trt=3;
    count=&_trt3;
    output;
    _trt=4;
    count=&_trt4;
    output;
run;

data _denomf7;
    _datasrt=1;
    set _bydat1(keep=);
    * All treatment groups ;
    _trt1=0;
    _trt2=0;
    _trt3=0;
    _trt4=0;
    * _CAT is the subgroup variable ;

```

```

    _cat=1;
    output;
run;

proc sql noprint;
    select put(nobs - delobs, 12.) into :_nobs from dictionary.tables
        where (libname="WORK" and memname="_DENOM7");
    select setting into :miss from dictionary.options where
        upcase(optname)="MISSING";
quit;

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

proc sql noprint;
    select count(distinct COVBLSTN) into :totexpv from _anal7;
    select distinct COVBLSTN into :expv1 - :expv3 from _anal7 order by COVBLSTN;
quit;

data _frame7;
    _datasrt=1;
    set _bydat1(keep=);
    _blcksrt=7;
    length COVBLSTN 8;
    _catLabl=" ";
    _trt=1;
    COVBLSTN=1;
    _catord=1;
    _cat=1;
    output;
    _trt=2;
    COVBLSTN=1;
    _catord=1;
    _cat=1;
    output;
    _trt=3;
    COVBLSTN=1;
    _catord=1;
    _cat=1;
    output;
    _trt=4;
    COVBLSTN=1;
    _catord=1;
    _cat=1;
    output;
    _catLabl=" ";
    _trt=1;
    COVBLSTN=2;
    _catord=2;
    _cat=1;
    output;

```

```

    _trt=2;
    COVBLSTN=2;
    _catord=2;
    _cat=1;
    output;
    _trt=3;
    COVBLSTN=2;
    _catord=2;
    _cat=1;
    output;
    _trt=4;
    COVBLSTN=2;
    _catord=2;
    _cat=1;
    output;
    _catLabl = " ";
    _trt = 1;
    COVBLSTN = 9998;
    _catord = 3;
    _cat = 1;
    output;
    _trt = 2;
    COVBLSTN = 9998;
    _catord = 3;
    _cat = 1;
    output;
    _trt = 3;
    COVBLSTN = 9998;
    _catord = 3;
    _cat = 1;
    output;
    _trt = 4;
    COVBLSTN = 9998;
    _catord = 3;
    _cat = 1;
    output;
run;

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

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

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

    if _inframe;

    if count=. then
        count=0;

```

```

run;

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

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

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

  if last.COVBLSTN;
run;

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

  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=7;
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="COVBLSTN ";
_vrlabel="Baseline SARS-CoV-2 status ";
_rwlabel=put(COVBLSTN, sars.);

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

if _catord=. then
    _catord=9997;

run;

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

data _base7;
    length _catlabl $200;
    set _pct7 end=eof;
    by _datasrt _blcksrt _catord COVBLSTN _trt _cat;
    retain _rowsrt 0 _rowmax 0;
    array _trtcnt(*) _trt1- _trt5;
    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;

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.COVBLSTN 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=3;
_dptindt=0;

if _trt=4 +1 then
    _trt=9999;

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

run;

options orientation=LANDSCAPE papersize="LETTER";
ods escapechar="~";
title1 "Demographic Characteristics (*ESC*){unicode 2013} Subjects 12 Through 15 and 16 Through 25 Years of Age
(*ESC*){unicode 2013} Safety Population";
footnote1
    "Abbreviation: SARS-CoV-2 = severe acute respiratory syndrome coronavirus 2.";
footnote2 "Note: Human immunodeficiency virus (HIV)-positive subjects are included in this summary but analyzed
and reported separately.";
footnote3 "a.(*ESC*){nbspspace 5}N = number of subjects in the specified group, or the total sample. This value is the
denominator for the percentage calculations.";
footnote4
    "b.(*ESC*){nbspspace 5}n = Number of subjects with the specified characteristic.";
footnote5 "c.(*ESC*){nbspspace 5}Positive N-binding antibody result at Visit 1, positive NAAT result at Visit 1, or
medical history of COVID-19.";

```

footnote6 "d.(*ESC*){nbspspace 5}Negative N-binding antibody result at Visit 1, negative NAAT result at Visit 1, and no medical history of COVID-19.";

```
data _final;
  set _base1 _base2 _base3 _base4 _base5 _base6 _base7;
run;

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

data _final;
  set _final;
  drop __trt;

  if _trt=9999 then
    __trt=4 + 1;
  else
    __trt=_trt;

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

  if _column=9999 then
    _column=4 + 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 '^' & _datatyp='data' then
    _vrlbtag=1;
end;

if first._rowsrt then
do;
    _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('_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;

proc sql noprint;
  create table rson as select distinct _trt, _column , _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 ,
    _datasrt, _blcksrt, _vrlabel;

quit;

data ADSL_S005_DEMO_PED_SAF;
  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=ADSL_S005_DEMO_PED_SAF;
  by _datasrt _trt _blcksrt _rowsrt;

run;

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

```

```

do start=1 to 4 + ("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_S005_DEMO_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 _blksrt _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 _blksrt _rowsrt _rvalue _trt;
run;

data _pre_transposed;
    set _pre_transposed;

    if _trt=9999 then
        _trt=4 +1;
run;

proc transpose data=_pre_transposed out=_column_transposed (drop=_name_)
    prefix=TRT;
    by _fixvar _fix2var _datasrt _blksrt _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_s005_demo_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}"
("BNT162b2 (30 (*ESC*){unicode 03BC}g)~{line}"
    TRT1 TRT2) ("Placebo~{line}" TRT3 TRT4) _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 "&_trlbl1." spacing=2 style(column)={width=35mm
        leftmargin=12px} style(header)={just=center} center;
    define TRT2 / group nozero "&_trlbl2." spacing=2 style(column)={width=35mm
        leftmargin=12px} style(header)={just=center} center;
    define TRT3 / group nozero "&_trlbl3." spacing=2 style(column)={width=35mm
        leftmargin=12px} style(header)={just=center} center;
    define TRT4 / group nozero "&_trlbl4." 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;

```