```
*************************************************************************************************
****************;
** Program Name   : adsl.sas                                                                    **;
** Date Created   : 17Nov2021                                                                   **;
** Programmer Name: (b) (4), (b) (6)                                                            **;
** Purpose        : Create adsl dataset                                                         **;
** Input data     : dm suppdm ex suppex ds suppds is co lb cm ie dv suppdv vs sv mb mh face ce ho suppho      **;
** External file  : ../prjC459/nda2_unblinded_esub/sbla1215_esub_adam/saseng/cdisc3_0/output/xpt            **;
** Output data    : adsl.sas7bdat                                                               **;
*************************************************************************************************
****************;
%let
oprot=/Volumes/app/cdars/prod/sites/cdars4/prjC459/nda2_unblinded_esub/sbla1215_esub_sdtm/saseng/cdisc3_0/data/
sdtm;
%let
protori=/Volumes/app/cdars/prod/sites/cdars4/prjC459/nda2_unblinded_esub/sbla1215_esub_adam/saseng/cdisc3_0;
%let
prot=/Volumes/app/cdars/prod/sites/cdars4/prjC459/nda2_unblinded_esub/sbla1215_esub_adam/saseng/cdisc3_0/analy
sis/eSUB;
*Path for external files;
%let
expath=/Volumes/app/cdars/prod/sites/cdars4/prjC459/nda2_unblinded_esub/sbla1215_esub_adam/saseng/cdisc3_0/out
put/xpt;

libname dataprot "&oprot." access=readonly;
libname datvprot "&protori./data_vai" access=readonly;
libname datvout "&prot./data_vai";

*Insert the date of snapshot;
%let cutoff2=02SEP2021;

proc printto print="&prot./output/adsl.rpt"
      log="&prot./logs/adsl.log" new;
run;
*************************************************************************************************;
* Clean *;
*************************************************************************************************;

proc delete data=work._all_;
run;

*************************************************************************************************;
* Format *;
*************************************************************************************************;

proc format;
   invalue sex "M"=1 "F"=2;
   invalue race "WHITE"=1 "BLACK OR AFRICAN AMERICAN"=2
      "AMERICAN INDIAN OR ALASKA NATIVE"=3 "ASIAN"=4
      "NATIVE HAWAIIAN OR OTHER PACIFIC ISLANDER"=5 "MULTIPLE"=6 "NOT REPORTED"=7;
   invalue ethnic "HISPANIC OR LATINO"=1 "NOT HISPANIC OR LATINO"=2
      "NOT REPORTED"=3 "UNKNOW"=4;
   invalue aethnic "HISPANIC OR LATINO"=1 "NOT HISPANIC OR LATINO"=2
```

```
      "NOT REPORTED"=3 "UNKNOW"=4;
   invalue arace "WHITE"=1 "BLACK OR AFRICAN AMERICAN"=2
      "AMERICAN INDIAN OR ALASKA NATIVE"=3 "ASIAN"=4
      "NATIVE HAWAIIAN OR OTHER PACIFIC ISLANDER"=5 "MULTIRACIAL"=6
      "NOT REPORTED"=7 "UNKNOWN"=8;
   invalue RaceGr1x "WHITE"=1 "BLACK OR AFRICAN AMERICAN"=2 "ALL OTHERS"=3;
   invalue RacialD "JAPANESE"=5 "OTHER"=999;
   invalue RANDAGE "12-15 Years"=1 "16-55 Years"=2 "18-55 Years"=3
      "65-85 Years"=4 ">55 Years"=5;
   invalue INFAGE "12-15 Years"=1 "16-55 Years"=2 "18-55 Years"=3 "65-85 Years"=4
      ">55 Years"=5;
   value $stat 'UNK'='1' 'UNKNOWN'='1' 'N'='2' 'NEG'='2' 'IND'='3' 'Y'='4'
      'POS'='4';
   value stat 0=' ' 1='UNK' 2='NEG' 3='IND' 4='POS';
   invalue trtfmt "BNT162b1 Phase 1 (10 mcg)"=1 "BNT162b1 Phase 1 (20 mcg)"=2
      "BNT162b1 Phase 1 (30 mcg)"=3 "BNT162b1 Phase 1 (100/10 mcg)"=4
      "BNT162b2 Phase 1 (10 mcg)"=5 "BNT162b2 Phase 1 (20 mcg)"=6
      "BNT162b2 Phase 1 (30 mcg)"=7 "BNT162b2 Phase 2/3 (30 mcg)"=8 "Placebo"=9
      "BNT162b2-Naive Participants - BNT162b2SA (30 mcg) 2 Doses"=10
      "BNT162b2 (30 mcg, 1 Dose)"=11 "BNT162b2SA (30 mcg, 1 Dose)"=12
      "BNT162b2SA (30 mcg, 2 Doses)"=13 "BNT162b2 (5 mcg, 1 Dose)"=14
      "BNT162b2 (10 mcg, 1 Dose)"=15 "Indeterminate"=98;
run;

*************************************************************************************************;
* Read in source DM/DS/EX SDTM datasets. *;
*************************************************************************************************;

Data DmSet;
   Set dataprot.dm;
Run;

Data DsSet;
   Set dataprot.ds;
Run;

Data ExSet;
   Set dataprot.Ex;
Run;

data prd2;
   set ExSet;

   if (index(visit, "_VAX3") or index(visit, "_VAX4") or
      visit='V303_MONTH1_POSTVAX3') and exstdtc ne "";

proc sort;
   by usubjid visitnum;
run;

proc sort data=prd2 nodupkey;
   by usubjid;
run;
```

```
proc sql UNDO_POLICY=NONE;
   create table ExSet as select a.*, case when not missing(b.exstdtc) and not
      missing(a.exstdtc) and .<input(scan(a.exstdtc, 1, "T"),
      yymmdd10.)<input(scan(b.exstdtc, 1, "T"), yymmdd10.) then
      "Double Blinded Period" when not missing(a.exstdtc) and missing(b.exstdtc)
      then "Double Blinded Period" else "Open Label Period" end as PERIOD, case
      when (calculated period)="Double Blinded Period" then 1 when (calculated
      period)="Open Label Period" then 2 end as PERIODN, case when not
      missing(a.extrt) and a.extrt ne "INDETERMINATE" then a.extrt else "ZZZZZZ"
      end as _extrt_ from ExSet a left join prd2 b on a.usubjid=b.usubjid order by
      periodn, period, usubjid, _extrt_;
quit;

data ExSet;
   set ExSet;
   by periodn period usubjid _extrt_;
   retain ACTDRUG;

   if first.usubjid then
      ACTDRUG=extrt;

   if index(ACTDRUG, "BNT")=0 then
      ACTDRCD=upcase(ACTDRUG);
   else if not missing(ACTDRUG) then
      ACTDRCD=substr(upcase(ACTDRUG), length(ACTDRUG)-1, 2);
run;

proc sort data=DsSet out=randcode nodupkey;
   by usubjid;
   where dsrefid ne "";
run;

proc sql undo_policy=none;
   create table DmSet as select a.*, b.dsrefid as tmpid, c.tmpdtc, d.qval as
      DSRANGRP from DmSet a left join randcode b on a.usubjid=b.usubjid left
      join (select distinct usubjid, max(exstdtc) as tmpdtc from ExSet where not
      missing(exstdtc)) c on a.usubjid=c.usubjid left join
      dataprot.suppds(where=(qnam="DSRANGRP")) d on a.usubjid=d.usubjid order by
      usubjid;
quit;

/*Assign dose level variables*/
data DmSet;
   set DmSet;
   length dosalvl dosplvl $100;
   label DOSPLVL="Planned Dosing Level" DOSPLVLN="Planned Dosing Level (N)"
      DOSALVL="Actual Dosing Level" DOSALVLN="Actual Dosing Level (N)";

   if tmpid ne "" then
      do;

         if armcd="PLACEBO" and not missing(dsrangrp) then
            do;
               dosplvl="Placebo";
```

```
            dosplvln=6;
         end;
      else
         do;
            _dosplvl=scan(scan(DSRANGRP, 3, ","), 1, " ");
            dosplvl=tranwrd(tranwrd(scan(scan(scan(DSRANGRP, 3, ","), 2, "("), 1,
               ")"), "mcg", " (*ESC*){unicode 03BC}g"), "100",
               "100 (*ESC*){unicode 03BC}g/10");

            if _dosplvl="TBD" then
               dosplvln=0;

            if _dosplvl="Low" then
               dosplvln=1;

            if _dosplvl="Low-Mid" then
               dosplvln=2;

            if _dosplvl="Medium" then
               dosplvln=3;

            if _dosplvl="Mid-High" then
               dosplvln=4;

            if _dosplvl="High" then
               dosplvln=5;
         end;

      if not missing(tmpdtc) then
         do;
            dosalvl=dosplvl;
            dosalvln=dosplvln;
         end;
   end;

  if missing(arm) then
   do;
      arm='BLINDED THERAPY';
      armcd='Z';
      actarm='BLINDED THERAPY';
      actarmcd='Z';
   end;
run;

*****************************************************************;
*Reading INPUT SDTM and Supplemental Datasets *;
*Merge DM and SUPPDM*;
*****************************************************************;

data _spmdel_supp_dsin_subset;
  set dataprot.suppdm;
run;

data _spmdel_sdtm_ds;
```

```
    set DmSet;
run;

data _spmdel_supp_dsin_subset_idvar1;
    set _spmdel_supp_dsin_subset;
    where idvar is missing;
run;

proc sort data=_spmdel_supp_dsin_subset_idvar1;
    by studyid usubjid idvar idvarval;
    quit;

proc transpose data=_spmdel_supp_dsin_subset_idvar1
        out=_spmdel_supp_dsin_idvar1_h;
    by studyid usubjid idvar idvarval;
    id qnam;
    idlabel qlabel;
    var qval;
    quit;

data _spmdel_suppds1 (drop=idvar idvarval _NAME_ _LABEL_);
    set _spmdel_supp_dsin_idvar1_h;

    if idvar='';
run;

proc sort data=_spmdel_sdtm_ds out=_ds1;
    by STUDYID USUBJID;
run;

proc sort data=_spmdel_suppds1 out=_ds2;
    by STUDYID USUBJID;
run;

data _spmdel_sdtm_temp_out1;
    merge _ds1(in=d1) _ds2(in=d2);
    by STUDYID USUBJID;

    if d1;
run;

data DmSet;
    set _spmdel_sdtm_temp_out1;
run;

*****************************************************************;
*Reading INPUT SDTM and Supplemental Datasets *;
*Merge EX and SUPPEX*;
*****************************************************************;

data _spmdel_supp_dsin_subset;
    set dataprot.suppex;
    where;
run;
```

```
data _spmdel_sdtm_ds;
   set ExSet;
run;

data _spmdel_supp_dsin_subset_idvar1;
   set _spmdel_supp_dsin_subset;
   where idvar="EXSEQ";
run;

proc sort data=_spmdel_supp_dsin_subset_idvar1;
   by studyid usubjid idvar idvarval;
   quit;

proc transpose data=_spmdel_supp_dsin_subset_idvar1
      out=_spmdel_supp_dsin_idvar1_h;
   by studyid usubjid idvar idvarval;
   id qnam;
   idlabel qlabel;
   var qval;
   quit;

data _spmdel_temp(keep=EXSEQ);
   set _spmdel_sdtm_ds;
run;

data _spmdel_suppds1 (drop=idvar idvarval _NAME_ _LABEL_);
   set _spmdel_supp_dsin_idvar1_h;

   if idvar="EXSEQ";
   EXSEQ=input(idvarval, best12.);
run;

proc sort data=_spmdel_sdtm_ds out=_ds1;
   by STUDYID USUBJID EXSEQ;
run;

proc sort data=_spmdel_suppds1 out=_ds2;
   by STUDYID USUBJID EXSEQ;
run;

data _spmdel_sdtm_temp_out1;
   merge _ds1(in=d1) _ds2(in=d2);
   by STUDYID USUBJID EXSEQ;

   if d1;
run;

data ExSet;
   set _spmdel_sdtm_temp_out1;
run;

*****************************************************************;
*Reading INPUT SDTM and Supplemental Datasets *;
```

```
*Merge DS and SUPPDS*;
************************************************************;

data _spmdel_supp_dsin_subset;
   set dataprot.suppds;
run;

data _spmdel_sdtm_ds;
   set DsSet;
run;

data _spmdel_supp_dsin_subset_idvar1;
   set _spmdel_supp_dsin_subset;
   where idvar="DSSEQ";
run;

proc sort data=_spmdel_supp_dsin_subset_idvar1;
   by studyid usubjid idvar idvarval;
   quit;

proc transpose data=_spmdel_supp_dsin_subset_idvar1
      out=_spmdel_supp_dsin_idvar1_h;
   by studyid usubjid idvar idvarval;
   id qnam;
   idlabel qlabel;
   var qval;
   quit;

data _spmdel_temp(keep=DSSEQ);
   set _spmdel_sdtm_ds;
run;

data _spmdel_suppds1 (drop=idvar idvarval _NAME_ _LABEL_);
   set _spmdel_supp_dsin_idvar1_h;

   if idvar="DSSEQ";
   DSSEQ=input(idvarval, best12.);
run;

proc sort data=_spmdel_sdtm_ds out=_ds1;
   by STUDYID USUBJID DSSEQ;
run;

proc sort data=_spmdel_suppds1 out=_ds2;
   by STUDYID USUBJID DSSEQ;
run;

data _spmdel_sdtm_temp_out1;
   merge _ds1(in=d1) _ds2(in=d2);
   by STUDYID USUBJID DSSEQ;

   if d1;
run;
```

```
data DsSet;
   set _spmdel_sdtm_temp_out1;
run;

*******************************************************************************;
* Derive the date/time variables. *;
*******************************************************************************;

proc sort data=ExSet out=_ds1;
   by UsubjId;
run;

proc sort data=DmSet out=_ds2;
   by UsubjId;
run;

data ExSet;
   merge _ds1(in=d1) _ds2(in=d2 drop=domain);
   by UsubjId;

   if d1;
run;

data ExSet;
   set ExSet;
   ExStDt=input(ExStDtc, ??is8601da.);
   format ExStDt date9.;
   ExEnDt=input(ExEnDtc, ??is8601da.);
   format ExEnDt date9.;
   ExStTm=.;

   if length(strip(ExStDtc))>12 then
      do;
         ExStTm=input(substr(ExStDtc, 12), ??is8601tm.);
      end;
   format ExStTm time8.;
   ExEnTm=.;

   if length(strip(ExEnDtc))>12 then
      do;
         ExEnTm=input(substr(ExEnDtc, 12), ??is8601tm.);
      end;
   format ExEnTm time8.;
run;

data DmSet;
   set DmSet;

   if ^missing(BrthDtc) then
      do;
         length yr $4 mm dd $2;
         yr=substr(BrthDtc, 1, 4);
         mm=substr(BrthDtc, 6, 2);
         dd=substr(BrthDtc, 9, 2);
```

```
      if yr ne ' ' then
         do;
            dflag=' ';

            if (dd eq "  " or dd eq "-T") and mm ne " " then
               do;
                  dd='01';
                  dflag='D';
               end;

            if mm eq "  " or mm eq "--" then
               do;
                  mm='01';
                  dd='01';
                  dflag='M';
               end;
            newdate=(trim(left(yr))||'-'||trim(left(mm))||'-'||trim(left(dd)));
            BrthDt=input(newdate, ??is8601da.);
            format BrthDt date9.;
            BrthDtF=dflag;
         end;
      drop yr mm dd dflag newdate;
   end;
RfxStDt=input(RfxStDtc, ??is8601da.);
format RfxStDt date9.;
RfxEnDt=input(RfxEnDtc, ??is8601da.);
format RfxEnDt date9.;
RfStDt=input(RfStDtc, ??is8601da.);
format RfStDt date9.;
RfEnDt=input(RfEnDtc, ??is8601da.);
format RfEnDt date9.;
RfPEnDt=input(RfPEnDtc, ??is8601da.);
format RfPEnDt date9.;
RfIcDt=input(RfIcDtc, ??is8601da.);
format RfIcDt date9.;
RfxStTm=.;

if length(strip(RfxStDtc))>12 then
   do;
      RfxStTm=input(substr(RfxStDtc, 12), ??is8601tm.);
   end;
format RfxStTm time8.;
RfxEnTm=.;

if length(strip(RfxEnDtc))>12 then
   do;
      RfxEnTm=input(substr(RfxEnDtc, 12), ??is8601tm.);
   end;
format RfxEnTm time8.;
RfStTm=.;

if length(strip(RfStDtc))>12 then
   do;
```

```
        RfStTm=input(substr(RfStDtc, 12), ??is8601tm.);
      end;
   format RfStTm time8.;
   RfEnTm=.;

   if length(strip(RfEnDtc))>12 then
      do;
        RfEnTm=input(substr(RfEnDtc, 12), ??is8601tm.);
      end;
   format RfEnTm time8.;
   ;
run;

data DsSet;
   Set DsSet;
   DsStDT=input(DsStDtc, ??is8601da.);
   format DsStDT date9.;
   DsDt=input(DsDtc, ??is8601da.);
   format DsDt date9.;
run;

*******************************************************************************************;
* Unique treatment group information *;
*******************************************************************************************;

Proc Sort Data=DmSet Out=UniqArm(Keep=Arm ArmCd) Nodupkey;
   By Arm Armcd;
   Where upcase(strip(ArmCd)) not in ("SCRNFAIL", "NOTASSGN", " ");
Run;

Proc Sort Data=ExSet Out=UniqTrtVar(Keep=EXTRT) Nodupkey;
   By EXTRT;
   Where upcase(strip(EXTRT)) not in (" ");
Run;

Data UniqTrtVar;
   Length _TrtPhase $20. _TrtVarOrd $20.;
   Set UniqTrtVar;
   _TrtVar=catx(" ", EXTRT);
   _TrtVarOrd=byte(64+_n_);
   _TrtPhase="ACTIVE";
Run;

Data UniqArm;
   Length _TrtPhase $20. _TrtVarOrd $20.;
   Set Uniqarm;
   _TrtPhase="RANDOM";
   _TrtVar=Arm;
   _TrtVarOrd=ArmCd;
Run;

Data UniqTrtVarArm;
   Length _TrtVar _TrtVar2 $200.;
   Set UniqTrtvar UniqArm;
```

```
   _TrtVar2=_TrtVar;
Run;

data TrtMapTxt;
   set UniqTrtVarArm;
run;

Proc Sort data=TrtMapTxt;
   By _TrtVar _TrtPhase;
Run;

*******************************************************************************************;
* Individual's exposure information *;
*******************************************************************************************;

Data ExSet;
   Set ExSet;
   _TrtGrpC=catx(" ", EXTRT);
Run;

Proc Sort data=ExSet Out=ExTmChk;
   By ExStTm ExEnTm;
   Where ^Missing(ExStTm) or ^Missing(ExEnTm);
Run;

Data ExSet(Where=(nmiss(ExStDt, ExEnDt)=0)) ExDtmMiss;
   Set ExSet;

   If Nmiss(ExStDt, ExEnDt, ExStTm, ExEnTm)>=1 then
      output ExDtmMiss;
   Output ExSet;
Run;

Proc Sql NoPrint;
   Create table ExTemp1 as Select a.*, b._TrtVarOrd as _TrtGrpD From ExSet as a
      Left Join UniqTrtVarArm(where=(upcase(strip(_TrtPhase))="ACTIVE")) as b On
      a._TrtGrpC=b._TrtVar;
Quit;

Proc Sql NoPrint;
   Create table ExTemp2 as Select a.*, b._TrtVar2, b._TrtVarOrd From ExTemp1 as a
      Left Join TrtMapTxt(where=(upcase(strip(_TrtPhase))="ACTIVE")) as b On
      a._TrtGrpC=b._TrtVar;
Quit;

Data ExSet;
   Set ExTemp2;
   _TrtGrpC=ACTDRUG;
   _TrtGrpD=ACTDRCD;
Run;

Proc Sql NoPrint;
   Create table TrtxxP_A_N as Select distinct _TrtGrpD From ExSet Order By
      _TrtGrpD;
```

```
Quit;

Data TrtxxP_A_N;
   Set TrtxxP_A_N;
   FmtName="TrtNFmt";
   Start=_TrtGrpD;
End=_TrtGrpD;
Label=_n_;
Type="I";
Run;

Proc Format Lib=Work CntlIn=TrtxxP_A_N;
Run;

Proc Sql NoPrint;
   Create table TrtGrpMac as Select distinct _TrtGrpD, _TrtGrpC From ExSet Order
      By _TrtGrpD;
Quit;

Proc Sort Data=ExSet Out=ExTrt(Keep=Usubjid ExStDt ExEnDt ExStTm ExEnTm
      _TrtGrpC _TrtGrpD) Nodupkey;
   By Usubjid ExStDt ExStTm ExEnDt ExEnTm _TrtGrpD _TrtGrpC;
Run;

Data ExTrt;
   Retain _TrtGrpN 0;
   Set ExTrt;
   By Usubjid _TrtGrpD _TrtGrpC NotSorted;

   If First._TrtGrpC then
      do;

         If First.Usubjid then
            _TrtGrpN=0;
         _TrtGrpN=_TrtGrpN+1;
      End;
run;

Proc Sort Data=ExTrt Out=ExTrtA;
   By Usubjid _TrtGrpN;
Run;

Proc Sort Data=ExTrt Out=ExTrtSt;
   By Usubjid _TrtGrpN ExStDt ExStTm;
Run;

Proc Sort Data=ExTrt Out=ExTrtEt;
   By Usubjid _TrtGrpN ExEnDt ExEnTm;
Run;

Data ExTrtA;
   Set ExTrtA;
   By UsubjId _TrtGrpN;
   _xx=Strip(Put(_TrtGrpN, z2.));
```

```
   If First._TrtGrpN then
      do;
         _TrtxxA=Cats("TRT", _xx, "A");
         Output ExTrtA;
      End;
Run;

Data ExTrtSt ExTrtSdt(Keep=Usubjid TrtSdt TrtStm);
   Set ExTrtSt;
   By UsubjId _TrtGrpN;
   _xx=Strip(Put(_TrtGrpN, z2.));

   If First._TrtGrpN then
      do;
         _TrxxSdt=Cats("TR", _xx, "SDT");
         _TrxxStm=Cats("TR", _xx, "STM");
         Output ExTrtSt;
      End;

   If First.UsubjId then
      do;
         TrtSdt=ExStDt;
         TrtStm=ExStTm;
         Output ExTrtSdt;
      End;
Run;

Data ExTrtEt ExTrtEdt(Keep=UsubjId TrtEdt TrtEtm);
   Set ExTrtEt;
   By UsubjId _TrtGrpN;
   _xx=Strip(Put(_TrtGrpN, z2.));

   If Last._TrtGrpN then
      do;
         _TrxxEdt=Cats("TR", _xx, "EDT");
         _TrxxEtm=Cats("TR", _xx, "ETM");
         Output ExTrtEt;
      End;

   If Last.UsubjId then
      do;
         TrtEdt=ExEnDt;
         TrtEtm=ExEnTm;
         Output ExTrtEdt;
      End;
Run;

Proc Transpose Data=ExTrtA Out=ExTrtxxA;
   By UsubjId;
   Id _TrtxxA;
   Var _TrtGrpC;
Run;
```

```
Proc Transpose Data=ExTrtA Out=ExTrtxxA2 Prefix=_ActArm;
   By UsubjId;
   Id _TrtGrpN;
   Var _TrtGrpC;
Run;

Proc Transpose Data=ExTrtA Out=ExTrtxxD Prefix=_ActArmCd;
   By UsubjId;
   Id _TrtGrpN;
   Var _TrtGrpD;
Run;

Proc Transpose Data=ExTrtSt Out=ExTrxxSdt;
   By UsubjId;
   Id _TrxxSdt;
   Var ExStDt;
Run;

Proc Transpose Data=ExTrtEt Out=ExTrxxEdt;
   By UsubjId;
   Id _TrxxEdt;
   Var ExEndt;
Run;

Proc Transpose Data=ExTrtSt Out=ExTrxxStm;
   By UsubjId;
   Id _TrxxStm;
   Var ExStTm;
Run;

Proc Transpose Data=ExTrtEt Out=ExTrxxEtm;
   By UsubjId;
   Id _TrxxEtm;
   Var ExEnTm;
Run;

Data ExSet_All;
   Merge ExTrtxxA(in=a) ExTrxxSdt ExTrxxStm ExTrxxEdt ExTrxxEtm ExTrtxxD
      ExTrtxxA2 ExTrtSdt ExTrtEdt;
   By UsubjId;

   if a;
   Derived_ActArm=strip(_ActArm1);
   Derived_ActArmCd=strip(_ActArmCd1);
   TR01SDTM=dhms(TR01SDT, 0, 0, TR01STM);
   TR01EDTM=dhms(TR01EDT, 0, 0, TR01ETM);
   TR02SDTM=dhms(TR02SDT, 0, 0, TR02STM);
   TR02EDTM=dhms(TR02EDT, 0, 0, TR02ETM);
   TrtSdtm=dhms(TrtSdt, 0, 0, TrtStm);
   TrtEdtm=dhms(TrtEdt, 0, 0, TrtEtm);
   Format TrtSdtm TrtEdtm datetime20.;
   Format TrtSdt TrtEdt date9. TrtStm TrtEtm time8. TR01SDTM TR01EDTM
      datetime20. TR02SDTM TR02EDTM datetime20.;
Run;
```

```
Data RfTimeMiss;
   Set DmSet;
   Where ^Missing(RfStTm) or ^Missing(RfEntm);
Run;

*****************************************************************************************;
* Individual's demographic information *;
*****************************************************************************************;

Proc Sql NoPrint;
   Create table DmSet_Adsl as Select a.*, b._TrtVar2 as Derived_Arm, b._TrtVarOrd
      as Derived_ArmCd From DmSet as a Left Join
      TrtMapTxt(where=(upcase(strip(_TrtPhase))="RANDOM")) as b On
      Upcase(Strip(a.Arm))=Upcase(Strip(b._TrtVar));
Quit;

*****************************************************************************************;
* Individual's disposition information *;
*****************************************************************************************;

Proc Sql NoPrint;
   Create table RandSet as Select distinct UsubjId, DsStDt as RandDt, DsRefId as
      RandNo From DsSet Where Upcase(Strip(DsDecod))="RANDOMIZED" and not
      missing(dsstdt) and not missing(dsrefid) Order by UsubjId, DsStDt;
Quit;

Data RandSet;
   Set RandSet;
   By UsubjId RandDt;

   If first.Usubjid;
Run;

Proc Sql NoPrint;
   Create table EnrlSet as Select distinct UsubjId, DsStDt as EnrlDt, DsRefId as
      EnrlNo From DsSet where index(upcase(strip(dsdecod)), "OBTAINED") and not
      missing(dsstdtc) Order by UsubjId, DsStDt;
Quit;

Data EnrlSet;
   Set EnrlSet;
   By UsubjId EnrlDt;

   If first.Usubjid;
Run;

Proc Sort Data=DsSet Out=CmpFlSet(Keep=UsubjId DsStDt Rename=(DsStDt=ComplDt))
      NoDupKey;
   By UsubjId;
   Where Upcase(Strip(DsCat))="DISPOSITION EVENT" and
      Upcase(Strip(DsDecod))="COMPLETED" and Upcase(Strip(Dsphase))="FOLLOW-UP";
Run;
```

```
Data EosSet(Keep=UsubjId EosDcDt _EosDcClDt EosDcRs);
   Set DsSet;
   Where Upcase(Strip(DsCat))="DISPOSITION EVENT" and
      Upcase(Strip(DsDecod))^="COMPLETED" and Upcase(Strip(Dsphase))="FOLLOW-UP";
   EosDcDt=DsStDt;
   EosDcRs=DsDecod;
   _EosDcClDt=DsDt;
Run;

Proc Sort Data=EosSet;
   By UsubjId EosDcDt;
Run;

Data EosSet;
   Set EosSet;
   By UsubjId EosDcDt;

   If Last.UsubjId;
Run;

Data EotSet(Keep=UsubjId EotDcDt _EotDcClDt EotDcRs);
   Set DsSet;
   Where Upcase(Strip(DsCat))="DISPOSITION EVENT" and
      Upcase(Strip(DsDecod))^="COMPLETED" and Index(Upcase(Strip(Dsphase)),
      "VACCINATION")>0;
   EotDcDt=DsStDt;
   EotDcRs=DsDecod;
   _EotDcClDt=DsDt;
Run;

Proc Sort Data=EotSet;
   By UsubjId EotDcDt;
Run;

Data EotSet;
   Set EotSet;
   By Usubjid EotDcDt;

   If Last.Usubjid;
Run;

proc sort data=EosSet out=_ds1;
   by UsubjId;
run;

proc sort data=EotSet out=_ds2;
   by UsubjId;
run;

data EosEotSet;
   merge _ds1(in=d1 keep=UsubjId EosDcDt _EosDcClDt EosDcRs) _ds2(in=d2
      keep=UsubjId EotDcDt _EotDcClDt EotDcRs);
   by UsubjId;
```

```
   if d1 or d2;
run;

Data EosEotSet;
   Set EosEotSet;
   EosDcDt=coalesce(EosDcDt, _EosDcClDt);
   EotDcDt=coalesce(EotDcDt, _EotDcClDt);
   Drop _:;
Run;

*******************************************************************************************;
* Rebuild ActArm Process *;
*******************************************************************************************;

Proc Sort data=DmSet_Adsl Out=DmArm_cd(keep=Derived_Armcd Derived_Arm) Nodupkey;
   By Derived_ArmCd Derived_Arm;
Run;

Proc Sql;
   Create table ExSet_Adsl as Select distinct a.*, b.Derived_Arm as
      Derived_ActArm2 From ExSet_All as a Left Join DmArm_Cd as b On
      a.Derived_ActArmCd=b.Derived_ArmCd;
Quit;

*******************************************************************************************;
* Get variables from Demog Exposure and Disposition datatsets. *;
* Derive Population Flags, Demog Decode/Code variables. *;
* Re-Derive Arm/ArmCd/ActArm/ActArmCd variables. *;
* Derive Planned and Actual Treatment Sequence variables. *;
* Derive Trt<nn>A/Trt<nn>AN/Trt<nn>P/Trt<nn>PN variables. *;
* Derive Treatment and Analysis Period Date, Time and DateTime variables. *;
*******************************************************************************************;

Proc Sort Data=DmSet_Adsl;
   By UsubjId;
Run;

Proc Sort Data=ExSet_Adsl;
   By UsubjId;
Run;

Data Adsl;
   Length Arm ActArm Aethnic Arace $200. ArmCd ActArmCd $20. TRT01A $200.
      TRT02A $200. RaceGr1x RaceGr1 $100.;
   Merge DmSet_Adsl(in=_dm_) ExSet_Adsl(in=_ex_) RandSet (in=_ras_)
      EnrlSet (in=_enr_) CmpFlSet (in=_cmp_) EosEotSet;
   By UsubjId;

   If _dm_;

   If _dm_ and _ex_ then
      SafFl="Y";
   Else if _dm_ and ^_ex_ then
      SafFl="N";
```

```
If _dm_ and _ras_ then
   RandFl="Y";
Else
   RandFl="N";

if RandFL="N" then
   SafFL="N";

If _dm_ and _enr_ then
   EnrlFl="Y";
Else
   EnrlFl="N";

If _dm_ and _cmp_ then
   ComplFl="Y";
Else
   ComplFl="N";
SexN=input(Sex, ??sex.);
RaceN=input(Race, ??Race.);

if ^missing(Ethnic) then
   Aethnic=Ethnic;
else
   Aethnic="UNKNOWN";
EthnicN=input(Ethnic, ??Ethnic.);
AethnicN=input(Aethnic, ??Aethnic.);
RacialDN=input(RacialD, ??RacialD.);
RaceOth=" ";

If ^Missing(Race) then
   do;

      if upcase(Race)="MULTIPLE" then
         ARace="MULTIRACIAL";
      else
         Arace=Race;
   end;
else If missing(Race) and upcase(Race1)="NOT REPORTED" then
   Arace=Race1;
Else If ^Missing(RaceOth) then
   Arace="OTHER";
Else
   Arace="UNKNOWN";
AraceN=input(Arace, ??Arace.);

If upcase(Race)="WHITE" then
   RACEGR1x='WHITE';
Else If upcase(Race)="BLACK OR AFRICAN AMERICAN" then
   RACEGR1x='BLACK OR AFRICAN AMERICAN';
Else If upcase(Race) not in ("WHITE" "BLACK OR AFRICAN AMERICAN") then
   RACEGR1x='ALL OTHERS';
RaceGr1=RaceGr1x;
RaceGr1N=input(RaceGr1, ??RaceGr1x.);
```

```
Arm=coalescec(strip(Derived_Arm), Arm);
ArmCd=coalescec(strip(Derived_ArmCd), ArmCd);

If Missing(Derived_ActArmCd) and ArmCd not in ("SCRNFAIL", "NOTASSGN") then
   do;
      ActArmCd="NOTTRT";
      ActArm="Not Treated";
   End;
Else if ^Missing(Derived_ActArmCd) and Derived_ActArmcd^=Armcd then
   do;

      If derived_actarmcd in ("B2_P23_30", "PLACEBO") then
         do;
            ActArmcd=derived_actarmcd;
            Actarm=derived_actarm2;
         End;
      Else if derived_actarmcd in ("B1" "B2") and substr(armcd, 1,
         2)=substr(derived_actarmcd, 1, 2) then
            do;
            ActArmcd=armcd;
            Actarm=arm;
         End;
      Else if derived_actarmcd in ("B1" "B2") and not missing(dsrangrp) then
         do;
            ActArmcd=strip(derived_actarmcd)||"_"||strip(scan(dosalvl, 1, "("));
            Actarm=strip(derived_actarm)||" Phase 1 ("||strip(scan(dosalvl, 1,
               "("))||" mcg)";
         End;
      Else if derived_actarmcd in ("B1" "B2") and missing(dsrangrp) then
         do;
            ActArmcd=strip(derived_actarmcd)||"_P23_30";
            Actarm=strip(derived_actarm)||" Phase 2/3 (30 mcg)";
         End;
      Else if derived_actarmcd in ("PLACEBO") then
         do;
            ActArmcd="PLACEBO";
            Actarm="Placebo";
         End;
      Else if derived_actarmcd='SA' and substr(armcd, 1,
         2)=substr(derived_actarmcd, 1, 2) then
            do;
            ActArmcd="SA_30";
            Actarm="BNT162b2-Naive Participants - BNT162b2SA (30 mcg) 2 Doses";
         End;
      Else if derived_actarmcd not in ("B2_P23_30", "PLACEBO") and not
         missing(derived_actarmcd) then
            do;
            ActArmCd="NOTTRT";
            ActArm="Not Treated";
         End;
      Else if derived_actarmcd not in ("B2_P23_30", "PLACEBO") then
         do;
            ActArmCd="UNPLAN";
            ActArm="Unplanned Treatment";
```

```
            End;
        End;
    Else if ^Missing(Derived_ActArmCd) then
        do;
            ActArmCd=Derived_ActArmCd;
            ActArm=Derived_ActArm2;
        End;
    TrtSeqA=Strip(Derived_ActArm);
    TRT01AN=input(_ActArmCd1, ?? TrtNFmt.);
    TRT02AN=input(_ActArmCd2, ?? TrtNFmt.);
    Format ComplDt RandDt EnrlDt date9.;
    Drop ComplFl ComplDt RaceOth RaceGr1x;
Run;

*****************************************************************************************;
* Derive AAge and AAgeU variables. *;
* Derive Age<x> and Age<x>U from AAge [where x=Y,M,W,D and H]. *;
* Derive Age Group related variables from Analysis Age variable [AAGE]. *;
*****************************************************************************************;

data adsl;
    set adsl;

    if not missing(RANDDT) then
        ENRLDT=RANDDT;
    else if not missing(RFICDTC) then
        ENRLDT=input(RFICDTC, yymmdd10.);
run;

data adsl;
    set adsl;
    length aageu $6 agegr1 $100 RANDAGE $100;
    _birthday=day(brthdt);
    _birthmth=month(brthdt);
    _birthyr=year(brthdt);
    _EnrlDtday=day(EnrlDt);
    _EnrlDtmth=month(EnrlDt);
    _EnrlDtyr=year(EnrlDt);
    aage=_EnrlDtyr - _birthyr;

    if (_EnrlDtmth lt _birthmth) or ((_EnrlDtmth eq _birthmth) and (_EnrlDtday lt
        _birthday)) then
            do;
            aage=aage - 1;
        end;

    if n(aage) then
        aageu="YEARS";

    If 12<=(aage)<=15 and missing(dsrangrp) then
        RANDAGE='12-15 Years';
    Else If 16<=(aage)<=55 and missing(dsrangrp) then
        RANDAGE='16-55 Years';
    Else If 18<=(aage)<=55 and not missing(dsrangrp) then
```

```
      RANDAGE='18-55 Years';
   Else If 65<=(aage) and not missing(dsrangrp) then
      RANDAGE='65-85 Years';
   Else If 56<=(aage) and missing(dsrangrp) then
      RANDAGE='>55 Years';
   agegr1=RANDAGE;
   agegr1N=input(agegr1, ??RANDAGE.);
   drop RANDAGE;
   ;
run;

data adsl;
   set adsl;
   length aageyu $6;
   _birthday=day(brthdt);
   _birthmth=month(brthdt);
   _birthyr=year(brthdt);
   _EnrlDtday=day(EnrlDt);
   _EnrlDtmth=month(EnrlDt);
   _EnrlDtyr=year(EnrlDt);
   aagey=_EnrlDtyr - _birthyr;

   if (_EnrlDtmth lt _birthmth) or ((_EnrlDtmth eq _birthmth) and (_EnrlDtday lt
      _birthday)) then
         do;
         aagey=aagey - 1;
      end;

   if n(aagey) then
      aageyu="YEARS";
   length aagemu $6;
   aagem=int((EnrlDt-brthdt+1)/30.4375);

   if n(aagem) then
      aagemu="MONTHS";
   length aagewu $6;
   aagew=int((EnrlDt-brthdt+1)/7);

   if n(aagew) then
      aagewu="WEEKS";
   length aagedu $6;
   aaged=EnrlDt - brthdt + 1;

   if n(aaged) then
      aagedu="DAYS";
   length aagehu $6;
   aageh=(EnrlDt - brthdt + 1)*24;

   if n(aageh) then
      aagehu="HOURS";
run;

*******************************************************************************************;
* Planned Treatments Sequence from Treatmap *;
```

```
*************************************************************************************************;

Data Adsl;
   Set Adsl;
   Length TrtSeqP $200. _TrtpTmp1 TRT01P $200.;
   _TrtpArmCd1=ArmCd;
   TRT01P=Arm;
   _TrtpTmp1=TRT01P;

   if _TrtpArmCd1="B2_P23_30" then
      TRT01PN=1;
   else if _TrtpArmCd1="PLACEBO" then
      TRT01PN=2;
   TrtSeqP=_TrtPTmp1;
Run;

*************************************************************************************************;
* Derive Vaccination Dates and Age related variables. *;
*************************************************************************************************;

proc sort data=exset out=exuse(rename=(visitnum=vstn_org visit=vst_org));
   by usubjid exstdtc exendtc visitnum;
   where visitnum ne .;
run;

data exset;
   set exuse;

   if index(vst_org, "_VAX1") then
      visitnum=1;

   if index(vst_org, "_VAX2") then
      visitnum=2;

   if index(vst_org, "_VAX3") then
      visitnum=3;

   if index(vst_org, "_VAX4") or vst_org='V303_MONTH1_POSTVAX3' then
      visitnum=4;

   if vst_org="" and period="Double Blinded Period" then
      visitnum=2.01;
   else if vst_org="" and period="Open Label Period" then
      visitnum=4.01;
run;

proc sort data=exset out=exvac_(keep=usubjid visitnum exstdt) nodupkey;
   by visitnum usubjid;
run;

data exvacxx1;
   set exvac_;
   where visitnum in (1 2 2.01);
run;
```

```
data exvacxx2;
   set exvac_;
   where visitnum in (3 4 4.01);
run;

data exvac;
   set exvacxx:;
run;

proc sort data=exvac out=exvac(keep=usubjid visitnum exstdt) nodupkey;
   by visitnum usubjid;
run;

proc sort data=exset out=exvisg1(keep=visitnum) nodupkey;
   by visitnum;
   where visitnum in (1 2 2.01);
run;

data exvisg1;
   set exvisg1 end=eof;
   by visitnum;
   length vaxvar $8 vaxlabel $40;
   vaxg=1;
   vaxn=put(_n_, z2.);
   vaxvar=cats("VAX", "1", vaxn, "DT");
   vaxlabel=ifC(1 eq 1, catx(" ", "Vaccination Date", vaxn), catx(" ",
      "Vaccination Group1 Date", vaxn));

   if eof then
      call symputx(cats('_nvax', 1), cats(_n_));
   ;
run;

data exvis(index=(visitnum));
   set exvisg1;
run;

proc sort data=exset out=exvisg2(keep=visitnum) nodupkey;
   by visitnum;
   where visitnum in (3 4 4.01);
run;

data exvisg2;
   set exvisg2 end=eof;
   by visitnum;
   length vaxvar $8 vaxlabel $40;
   vaxg=2;
   vaxn=put(_n_, z2.);
   vaxvar=cats("VAX", "2", vaxn, "DT");
   vaxlabel=ifC(2 eq 1, catx(" ", "Vaccination Date", vaxn), catx(" ",
      "Vaccination Group2 Date", vaxn));

   if eof then
```

```
      call symputx(cats('_nvax', 2), cats(_n_));
   ;
run;

data exvis(index=(visitnum));
   set exvis exvisg2;
run;

data rfadsl(keep=usubjid brthdt index=(usubjid));
   set adsl;
run;

data exvac1;
   set exvac;
   set exvis(keep=visitnum vaxg vaxn vaxvar vaxlabel) key=visitnum/unique;
   set rfadsl key=usubjid/unique;

   if _error_=1 then
      do;
         _error_=0;
         call missing(brthdt);
      end;
   cvalue=exstdt;
   output exvac1;

   if vaxg=1 then
      do;
         vaxvar=cats('AGETR', vaxn);
         vaxlabel=catx(' ', 'Age at Vaccination', vaxn);
         _birthday=day(brthdt);
         _birthmth=month(brthdt);
         _birthyr=year(brthdt);
         _exstdtday=day(exstdt);
         _exstdtmth=month(exstdt);
         _exstdtyr=year(exstdt);
         cvalue=_exstdtyr - _birthyr;

         if (_exstdtmth lt _birthmth) or ((_exstdtmth eq _birthmth) and (_exstdtday
            lt _birthday)) then
               do;
               cvalue=cvalue - 1;
            end;
         output exvac1;
      end;
run;

proc sort data=exvac1;
   by usubjid;
run;

proc transpose data=exvac1 out=exvac2(drop=_:);
   by usubjid;
   id vaxvar;
   idlabel vaxlabel;
```

```
    var cvalue;
run;

data exvac2;
    set exvac2;
    attrib agetru01 label="Age Units at Vaccination 01" length=$6;

    if n(agetr01) then
       agetru01="YEARS";
    format vax101dt date9.;
    attrib agetru02 label="Age Units at Vaccination 02" length=$6;

    if n(agetr02) then
       agetru02="YEARS";
    format vax102dt date9.;
    attrib agetru03 label="Age Units at Vaccination 03" length=$6;

    if n(agetr03) then
       agetru03="YEARS";
    format vax103dt date9.;
    format vax201dt date9.;
    format vax202dt date9.;
    length INFAGE agetgr1 $40.;

    If 12<=(agetr01)<=15 and missing(dsrangrp) then
       INFAGE='12-15 Years';
    Else If 16<=(agetr01)<=55 and missing(dsrangrp) then
       INFAGE='16-55 Years';
    Else If 18<=(agetr01)<=55 and not missing(dsrangrp) then
       INFAGE='18-55 Years';
    Else If 65<=(agetr01) and not missing(dsrangrp) then
       INFAGE='65-85 Years';
    Else If 56<=(agetr01) and missing(dsrangrp) then
       INFAGE='>55 Years';
    agetgr1=INFAGE;
    agetgr1n=input(INFAGE, ??INFAGE.);
    attrib agetgr1 label="Age Group at Vaccination 01" agetgr1n
       label="Age Group at Vaccination 01 (N)";
run;

data adsl;
    merge adsl(in=a) exvac2(keep=usubjid vax101dt vax102dt vax103dt vax201dt
       vax202dt agetr01 agetru01 agetr02 agetru02 agetr03 agetru03 agetgr1n agetgr1);
    by usubjid;

    if a;
run;

*******************************************************************************;
* VS *;
*******************************************************************************;

data srv_vs;
    set dataprot.vs;
```

```
    vsdt=input(vsdtc, ??is8601da.);
    format vsdt date9.;
    keep usubjid vsdt;
    where ^missing(vsorres);
run;

proc sort data=srv_vs;
    by usubjid vsdt;
run;

data srv_vs;
    set srv_vs;
    by usubjid vsdt;

    if last.usubjid;
run;

*******************************************************************************************;
* LB *;
*******************************************************************************************;

data srv_lb;
    set dataprot.lb;
    lbdt=input(lbdtc, ??is8601da.);
    format lbdt date9.;
    keep usubjid lbdt;
    where ^missing(lborres);
run;

proc sort data=srv_lb;
    by usubjid lbdt;
run;

data srv_lb;
    set srv_lb;
    by usubjid lbdt;

    if last.usubjid;
run;

*******************************************************************************************;
* CM *;
*******************************************************************************************;

data srv_cm;
    set dataprot.cm;
    cmstdt=input(cmstdtc, ??is8601da.);
    format cmstdt date9.;
    cmendt=input(cmendtc, ??is8601da.);
    format cmendt date9.;
    _maxcmdt=max(cmstdt, cmendt);
    keep usubjid _maxcmdt;
run;
```

```
proc sort data=srv_cm;
   by usubjid _maxcmdt;
run;

data srv_cm;
   set srv_cm;
   by usubjid _maxcmdt;

   if last.usubjid;
run;

*******************************************************************************************;
* DS *;
*******************************************************************************************;

data srv_ds;
   set dataprot.ds;
   dsstdt=input(dsstdtc, ??is8601da.);
   format dsstdt date9.;
   where dsdecod not in ("LOST TO FOLLOW-UP", "DEATH", "ENROLLED");
   keep usubjid dsstdt;
run;

proc sort data=srv_ds;
   by usubjid dsstdt;
run;

data srv_ds;
   set srv_ds;
   by usubjid dsstdt;

   if last.usubjid;
run;

*******************************************************************************************;
* DM *;
*******************************************************************************************;

data srv_dmadthdt;
   set dataprot.dm;

   if ^missing(dthdtc) then
      do;
         length yr $4 mm dd $2;
         yr=substr(dthdtc, 1, 4);
         mm=substr(dthdtc, 6, 2);
         dd=substr(dthdtc, 9, 2);
         ;

         if yr ne ' ' then
            do;
               dflag=' ';

               if (dd eq "  " or dd eq "-T") and mm ne " " then
```

```
                do;
                   dd='01';
                   dflag='D';
                end;

             if mm eq "  " or mm eq "--" then
                do;
                   mm='01';
                   dd='01';
                   dflag='M';
                end;
             newdate=(trim(left(yr))||'-'||trim(left(mm))||'-'||trim(left(dd)));
             adthdt=input(newdate, ??is8601da.);
             format adthdt date9.;
             adthdtF=dflag;
          end;
       drop yr mm dd dflag newdate;
    end;
  keep usubjid adthdt adthdtf;
run;

proc sort data=srv_dmadthdt;
  by usubjid;
run;

*************************************************************************************;
* Get recent most(max) from all assessment dates and along with Treatment Start/End and *;
* Randomization Date. *;
* Note: If derived ADTHDT is on or prior to SRVLACDT then reset as Last Contact Date +1.*;
*************************************************************************************;

data SrvSet;
  merge Adsl(in=a keep=usubjid trtsdt trtedt RANDDT dthfl) srv_dmadthdt srv_vs
     srv_lb srv_cm srv_ds;
  by usubjid;

  if a;
  _srvlacdt=max(trtsdt, trtedt, RANDDT, vsdt, lbdt, _maxcmdt, dsstdt);

  if ((^missing(adthdtf) and adthdt<=_srvlacdt) or (dthfl="Y" and
     missing(adthdt))) then
        adthdt=_srvlacdt+1;

  if missing(adthdt) then
     srvlacdt=_srvlacdt;
  attrib adthdt label="Analysis Date of Death" format=date9.;
  attrib srvlacdt label="Date of Last Contact" format=date9.;
  keep usubjid srvlacdt adthdt adthdtf;
run;

data adsl;
  merge adsl(in=a) srvset(in=b keep=usubjid adthdt: srvlacdt);
  by usubjid;
```

```
    if a;
    dthdt=adthdt;
    dthdtf=adthdtf;
    drop adthdt adthdtf;
    format dthdt date9.;
run;

*******************************************************************************************;
* Specification 1 *;
* ADD INDIVIDUAL BASELINE INFO. *;
* 1 - Cohort info. *;
* 2 - Phase info. *;
* 3 - Age Group variables. *;
* 4 - Pop for VOC in naive subjects *;
*******************************************************************************************;
*Cohort info;

data suppds;
    set dataprot.suppds;
    where qnam="DSRANGRP";

proc sort;
    by usubjid;
run;

data adsl;
    merge adsl(in=a) suppds(keep=usubjid qval rename=(qval=COHORT));
    by usubjid;
    label COHORT="Cohort Group" COHORTN="Cohort Group (N)";

    if index(cohort, "Stage 1") then
        srt1=1;
    else if index(cohort, "Stage 2") then
        srt1=2;

    if index(cohort, "21 Day") then
        srt2=1;
    else if index(cohort, "1-dose") then
        srt2=2;
    else if index(cohort, "60 Day") then
        srt2=3;

    if index(cohort, "Age 18 to 55") then
        srt3=1;
    else if index(cohort, "Age 65 to 85") then
        srt3=2;
    else if index(cohort, "Age 56 to 85") then
        srt3=3;

    if index(cohort, "BNT162a1") then
        srt4=1;
    else if index(cohort, "BNT162b1") then
        srt4=2;
    else if index(cohort, "BNT162b2") then
```

```
      srt4=3;
   else if index(cohort, "BNT162c2") then
      srt4=4;

   if index(cohort, "Low-") then
      srt5=2;
   else if index(cohort, "Low") then
      srt5=1;
   else if index(cohort, "-High") then
      srt5=4;
   else if index(cohort, "Medium") then
      srt5=3;
   else if index(cohort, "High") then
      srt5=5;

   if srt1=1 then
      do;

         if srt2=1 then
            cohortn=srt1+0.1+(srt3-1)*0.2+(srt4-1)*0.03+0.01+(srt5-1)*0.005;
         else if srt2>1 then
            cohortn=srt1+0.5+(srt3-1)*0.2+(srt2-2)*0.1+srt4*0.01;
      end;
   else if srt1>1 then
      cohortn=srt1+0.1+(srt3-1)/2*0.3+(srt2-1)*0.1+srt4*0.01+srt5*0.001;
   cohortn=round(cohortn, 0.001);
run;

*Distinct subject from phase 2 & 3;

proc import datafile="&expath./first-c4591001-360-participants-enrolled-v1-13aug20-update.xlsx"
      out=phase2 dbms=xlsx replace;
   getnames=no;
   datarow=2;
   sheet="SubjID_360";
run;

data phase2;
   set phase2;
   length subjid $20;
   subjid=B;

proc sort;
   by subjid;
run;

proc import
      datafile="&expath./newlist-c4591001-6k-participants-enrolled-v3-17sep2020.xlsx"
      out=phase3(rename=(a=var1)) dbms=xlsx replace;
   getnames=no;
   datarow=2;
run;

data phase3;
```

```
    set phase3;
    length subjid $20;
    subjid=scan(var1, -1, " ");

proc sort;
    by subjid;
run;

data adsl;
    merge adsl(in=a) /*phase1(keep=subjid in=b)*/
    phase2(keep=subjid in=c) phase3(keep=subjid in=d);
    by subjid;

    if a;
    attrib PHASEN label="Study Phase (N)" PHASE label="Study Phase" format=$200.;

    /*if b then
    do;
    PHASEN=1;
    PHASE="Phase 1";
    end;
    else */if c then
        do;
            PHASEN=2;
            PHASE="Phase 2_ds360/ds6000";
        end;
    else if d then
        do;
            PHASEN=3;
            PHASE="Phase 3_ds6000";
        end;
    else
        do;
            PHASEN=4;
            PHASE="Phase 3";
        end;
    dosplvl="";
    dosalvl="";
    dosplvln=.;
    dosalvln=.;
run;

*Categorize age groups;

data adsl;
    set adsl;

    if missing(AGETR01) then
        do;
            AGETR01=AAGE;
            AGETRU01=AAGEU;
            AGETGR1=AGEGR1;
            AGETGR1N=AGEGR1N;
        end;
```

```
AGEGR1=AGETGR1;
AGEGR1N=AGETGR1N;
length AGEGR2 AGEGR3 AGEGR4 $100.;

if PHASEN^=1 then
   do;

      if AGEGR1N=3 then
         do;
            AGEGR1N=2;
            AGEGR1="16-55 Years";
         end;

      if AGEGR1N=4 then
         do;
            AGEGR1N=5;
            AGEGR1=">55 Years";
         end;

      if 12<=AGETR01<=15 then
         do;
            AGEGR4="12-15 Years";
            AGEGR4N=1;
         end;

      if 16<=AGETR01<=25 then
         do;
            AGEGR4="16-25 Years";
            AGEGR4N=2;
         end;
   end;
else
   do;

      if AGEGR1N=2 then
         do;
            AGEGR1N=3;
            AGEGR1="18-55 Years";
         end;

      if AGEGR1N=5 then
         do;
            AGEGR1N=4;
            AGEGR1="65-85 Years";
         end;
   end;

if 65<=AGETR01 then
   do;
      AGEGR2N=2;
      AGEGR2=">=65 Years";
   end;
else if .<AGETR01<65 then
   do;
```

```
      AGEGR2N=1;
      AGEGR2="<65 Years";
    end;

  if 16<=AGETR01<=17 then
    do;
      AGEGR3N=1;
      AGEGR3="16-17 Years";
    end;
  else if 18<=AGETR01<=55 then
    do;
      AGEGR3N=2;
      AGEGR3="18-55 Years";
    end;
  else if AGETR01>55 then
    do;
      AGEGR3N=3;
      AGEGR3=">55 Years";
    end;

  if vax201dt>. and brthdtc ne "" then
    do;
      label agetr03="Age at Vaccination 03" agetru03="Age Units at Vaccination 03"
        agetgr3="Age Group at Vaccination 03"
        agetgr3n="Age Group at Vaccination 03 (N)";
      agetr03=floor((vax201dt-brthdt)/365.25);

      if substr(brthdtc, 5)=substr(strip(put(vax201dt, yymmdd10.)), 5) then
        agetr03=input(substr(strip(put(vax201dt, yymmdd10.)), 1, 4),
          best.)-input(substr(brthdtc, 1, 4), best.);
      agetru03="YEARS";

      if 18<=agetr03<=55 then
        do;
          agetgr3="18-55 Years";
          agetgr3N=1;
        end;
      else if agetr03>55 then
        do;
          agetgr3=">55 Years";
          agetgr3N=2;
        end;
    end;
  else
    do;
      agetr03=.;
      agetru03="";
    end;
run;

*****************************************************************************************;
* Specification 2 *;
* ADD PERIOD 2 TRT INFO AND VAX-SPECIFIC VARS *;
* 1 - VAXn0nTM/VAXn0n. *;
```

```
* 2 - TRT02P/TRT02PN. *;
* 3 - TR01:/TR02: for subjects that can not be distinguish as 2 periods *;
*************************************************************************************;
*VAXn0nTM/VAXn0n;

proc sql;
   create table ex as select a.*, b.phasen from ExSet a left join adsl b on
      a.usubjid=b.usubjid order by periodn, usubjid, visitnum;
   run;

proc sort data=ex nodupkey;
   by periodn usubjid visitnum;
run;

data ex;
   set ex;
   by periodn usubjid visitnum;

   if extptref in ("VACCINATION 1" "VACCINATION 3") then
      extmp=1;

   if extptref in ("VACCINATION 2" "VACCINATION 4") then
      extmp=2;

   if extptref="UNPLANNED VACCINATION" then
      extmp=3;
   extptref1n=periodn*100+extmp;
   extptref1="VAX"||strip(put(extptref1n, best.));
   format extime time8. exdatetime datetime20.;
   extime=input(scan(exstdtc, 2, "T"), time8.);
   exdatetime=input(exstdtc, is8601dt.);

   if exdosu in ("ug" "mcg") and index(extrt, "BNT") then
      extrt1=strip(extrt)||" ("||strip(exdose)||" (*ESC*){unicode 03BC}g)";
   else if phasen=1 and periodn=1 and index(extrt, "BNT") and index(dosalvl, "/")
      then
         extrt1=strip(extrt)||" ("||strip(scan(dosalvl, 1, "/"))||")";
   else if phasen=1 and periodn=1 and index(extrt, "BNT") and not
      missing(dosalvl) then
         extrt1=strip(extrt)||" ("||strip(dosalvl)||")";
   else if not (phasen=1 and periodn=1) and index(extrt, "BNT") then
      extrt1=strip(extrt)||" (30 (*ESC*){unicode 03BC}g)";
   else if upcase(extrt)='INDETERMINATE' then
      extrt1='Indeterminate';

   if extrt="Placebo" then
      extrt1=extrt;

proc sort;
   by usubjid extptref1n;
run;

proc transpose data=ex out=t_ex;
   by usubjid;
```

```
    id extptref1;
    var extrt1;
run;

proc transpose data=ex out=t_ex1 prefix=vax suffix=tm;
    by usubjid;
    id extptref1n;
    var extime;
run;

proc transpose data=ex out=t_ex2 prefix=vax suffix=dtm;
    by usubjid;
    id extptref1n;
    var exdatetime;
run;

*Correct Period 2 info;

data adsl(rename=(VAX103=VAX10U VAX203=VAX20U VAX103DT=VAX10UDT
        VAX203DT=VAX20UDT VAX103TM=VAX10UTM VAX203TM=VAX20UTM));
    attrib TR02EDT label='Date of Last Exposure in Period 02' TR02EDTM
        label='Datetime of Last Exposure in Period 02' TR02ETM
        label='Time of Last Exposure in Period 02' TR02SDT
        label='Date of First Exposure in Period 02' TR02SDTM
        label='Datetime of First Exposure in Period 02' TR02STM
        label='Time of First Exposure in Period 02' TRT02A
        label='Actual Treatment for Period 02' TRT02AN
        label='Actual Treatment for Period 02 (N)' VAX202 length=$200. VAX103DT
        format=date9. VAX202DT format=date9. VAX203DT format=date9. VAX103
        length=$200. VAX203 length=$200. VAX101TM length=8. format=time8. VAX102TM
        length=8. format=time8. VAX103TM length=8. format=time8. VAX201TM length=8.
        format=time8. VAX202TM length=8. format=time8. VAX203TM length=8.
        format=time8.;
    merge adsl(in=a drop=trtseqp trtseqa rename=(trt01p=_trt01p trt01pn=_trt01pn
        trt01a=_trt01a trt01an=_trt01an trt02a=_trt02a trt02an=_trt02an))
        t_ex(drop=_NAME_) t_ex1(drop=_NAME_) t_ex2(drop=_NAME_);
    by usubjid;

    if a;
    label VAX103DT="Vaccination Date Unplanned" VAX201DT="Vaccination Date 03"
        VAX202DT="Vaccination Date 04"
        VAX203DT="Vaccination Date Unplanned in Period 02" VAX101="Vaccination 01"
        VAX102="Vaccination 02" VAX103="Vaccination Unplanned"
        VAX201="Vaccination 03" VAX202="Vaccination 04"
        VAX203="Vaccination Unplanned in Period 02" VAX101TM="Vaccination Time 01"
        VAX102TM="Vaccination Time 02" VAX103TM="Vaccination Time Unplanned"
        VAX201TM="Vaccination Time 03" VAX202TM="Vaccination Time 04"
        VAX203TM="Vaccination Time Unplanned in Period 02";

    if TR01SDT ne . and (VAX101DT eq . and VAX102DT eq . and VAX103DT eq .) then
        do;
            TR01SDT=.;
            TR01SDTM=.;
            TR01STM=.;
```

```
      _trt01a='';
   end;

if TR02SDT ne VAX201DT and VAX201DT>. then
   do;
      TR01EDTM=max(VAX103DTM, VAX102DTM, VAX101DTM);
      TR01EDT=datepart(TR01EDTM);
      TR01ETM=timepart(TR01EDTM);
      TR02SDTM=VAX201DTM;
      TR02SDT=datepart(TR02SDTM);
      TR02STM=timepart(TR02SDTM);
      TR02EDTM=max(VAX203DTM, VAX202DTM, VAX201DTM);
      TR02EDT=datepart(TR02EDTM);
      TR02ETM=timepart(TR02EDTM);
   end;
length TRT01P TRT01A TRT02P TRT02A $200;

if arm in ("SCREEN FAILURE" "NOT ASSIGNED") then
   do;
      actarm=arm;
      actarmcd=armcd;
   end;

if _TRT01P not in ("SCREEN FAILURE" "NOT ASSIGNED") then
   do;

      if arm='BNT162b2-Naive Participants - BNT162b2SA (30 mcg) 2 Doses' then
         _trt01p='BNT162b2-Naive Participants - BNT162b2SA (30 mcg) 2 Doses';
      else if index(arm, '-') and index(arm, 'booster') then
         _trt01p='BNT162b2 Phase 2/3 (30 mcg)';
      TRT01P=_TRT01P;
      TRT01PN=input(TRT01P, trtfmt.);
   end;

if _TRT01A ne "" then
   do;

      if _TRT01A="Placebo" then
         TRT01A=_TRT01A;
      else if _TRT01A='BNT162b2SA' then
         TRT01A='BNT162b2-Naive Participants - BNT162b2SA (30 mcg) 2 Doses';
      else if PHASEN=1 and index(_TRT01A, "BNT") and index(dosalvl, "/")=0 then
         TRT01A=strip(_TRT01A)||" Phase 1 ("||tranwrd(scan(strip(dosalvl), 1, "/"),
            "(*ESC*){unicode 03BC}g", "mcg")||")";
      else if PHASEN=1 and index(_TRT01A, "BNT") then
         TRT01A=strip(_TRT01A)||" Phase 1 ("||strip(scan(scan(dosalvl, 1, "/"), 1,
            "(*ESC*)"))||"/"||strip(scan(scan(dosalvl, 2, "/"), 1,
            "(*ESC*)"))||" mcg)";
      else if index(_TRT01A, "BNT") then
         TRT01A=strip(_TRT01A)||" Phase 2/3 (30 mcg)";
      else if index(upcase(compress(vax101)||compress(vax102)||compress(vax103)),
         'INDETERMINATE') and
         index(upcase(compress(vax101)||compress(vax102)||compress(vax103)), 'BNT')
         eq 0 and
```

```
        index(upcase(compress(vax101)||compress(vax102)||compress(vax103)),
        'PLACEBO') eq 0 then
           TRT01A='Indeterminate';
   end;
TRT01AN=input(TRT01A, trtfmt.);

if index(arm, 'booster') eq 0 and arm not in ("SCREEN FAILURE" "NOT ASSIGNED")
   then
      do;

      if missing(trt01a) then
         do;
            actarm='Not Treated';
            actarmcd='NOTTRT';
         end;
   end;

if arm='BNT162b2-Naive Participants - BNT162b2SA (30 mcg) 2 Doses' then
   do;
      trt02p='';
   end;
else if tr02sdt>. and index(arm, '-')=0 then
   do;

      if PHASEN=1 then
         do;

            if trt01a='Placebo' then
               trt02p="BNT162b2 Phase 1 (30 mcg)";
            else if missing(trt01a) or trt01a='Indeterminate' then
               trt02p="BNT162b2 Phase 1 (30 mcg)";
            else if index(trt01a, '100') eq 0 and index(trt01a, 'BNT') then
               trt02p="BNT162b2 (30 mcg, 1 Dose)";
            else
               trt02p='';
         end;
      else
         do;

            if trt01a='Placebo' then
               trt02p="BNT162b2 Phase 2/3 (30 mcg)";
            else if missing(trt01a) or trt01a='Indeterminate' then
               trt02p="BNT162b2 Phase 2/3 (30 mcg)";
            else
               trt02p="";
         end;
   end;
else if index(arm, '-')then
   do;

      if strip(scan(arm, 2, '-'))='BNT162b2 (10 mcg) 1 booster' then
         trt02p='BNT162b2 (10 mcg, 1 Dose)';

      if strip(scan(arm, 2, '-'))='BNT162b2 (30 mcg) 1 booster' then
```

```
        trt02p='BNT162b2 (30 mcg, 1 Dose)';

      if strip(scan(arm, 2, '-'))='BNT162b2 (5 mcg) 1 booster' then
        trt02p='BNT162b2 (5 mcg, 1 Dose)';

      if strip(scan(arm, 2, '-'))='BNT162b2SA (30 mcg) 1 booster' then
        trt02p='BNT162b2SA (30 mcg, 1 Dose)';

      if strip(scan(arm, 2, '-'))='BNT162b2SA (30 mcg) 2 booster' then
        trt02p='BNT162b2SA (30 mcg, 2 Doses)';
    end;

if tr02sdt>. then
    do;

      if PHASEN=1 then
        do;

          if missing(trt01a) and (index(VAX201, "BNT") or index(VAX202, "BNT") or
            index(VAX203, "BNT")) then
              TRT02A="BNT162b2 Phase 1 (30 mcg)";
          ELSE if TRT01A='Placebo' and (index(VAX201, "BNT") or index(VAX202, "BNT")
            or index(VAX203, "BNT")) then
              TRT02A="BNT162b2 Phase 1 (30 mcg)";
        end;
      else
        do;

          if missing(trt01a) and (index(VAX201, "BNT") or index(VAX202, "BNT") or
            index(VAX203, "BNT")) then
              TRT02A="BNT162b2 Phase 2/3 (30 mcg)";
          else if TRT01A='Placebo' and (index(VAX201, "BNT") or index(VAX202, "BNT")
            or index(VAX203, "BNT")) then
              TRT02A="BNT162b2 Phase 2/3 (30 mcg)";
          else if index(vax201, 'BNT') and index(vax201, '30') and index(vax201,
            'SA') and index(trt02p, '1 Dose') ge 1 then
              TRT02A="BNT162b2SA (30 mcg, 1 Dose)";
          else if index(vax201, 'BNT') and index(vax201, '30') and index(vax201,
            'SA') and index(trt02p, '2 Doses') ge 1 then
              TRT02A="BNT162b2SA (30 mcg, 2 Doses)";
          else if index(vax201, 'BNT') and index(vax201, '5') and index(vax201,
            'SA') eq 0 then
              TRT02A="BNT162b2 (5 mcg, 1 Dose)";
          else if index(vax201, 'BNT') and index(vax201, '10') and index(vax201,
            'SA') eq 0 then
              TRT02A="BNT162b2 (10 mcg, 1 Dose)";
          else if index(vax201, 'BNT') and index(vax201, '30') and index(vax201,
            'SA') eq 0 then
              TRT02A="BNT162b2 (30 mcg, 1 Dose)";
        end;
    end;
else
    TRT02A='';
```

```
if index(upcase(compress(vax201)||compress(vax202)||compress(vax203)),
   'INDETERMINATE') and
   index(upcase(compress(vax201)||compress(vax202)||compress(vax203)), 'BNT') eq
   0 and index(upcase(compress(vax201)||compress(vax202)||compress(vax203)),
   'PLACEBO') EQ 0 then
      TRT02A='Indeterminate';

if not missing(trt01a) and missing(trt02a) and index(arm, '-') then
   do;

      if trt01a=trt01p then
         do;

            if arm='BNT162b2-Naive Participants - BNT162b2SA (30 mcg) 2 Doses' then
               do;
                  actarm=arm;
                  actarmcd=armcd;
               end;
            else
               do;
                  actarm=strip(scan(arm, 1, '-'));
                  actarmcd=strip(scan(armcd, 1, '-'));
               end;
         end;
      else
         do;
            *navie group and re-rand group will not take placebo for period 1;
            *IND will be considered;
         end;
   end;
else if not missing(trt01a) and not missing(trt02a) and index(arm, 'booster')
   then
      do;

      if trt01a='BNT162b2 Phase 2/3 (30 mcg)' and
         trt02a='BNT162b2 (30 mcg, 1 Dose)' then
            do;
            actarmcd='B2_P23_30-B2_30_1DB';
            actarm='BNT162b2 Phase 2/3 (30 mcg) - BNT162b2 (30 mcg) 1 booster';
         end;

      if trt01a='BNT162b2 Phase 2/3 (30 mcg)' and
         trt02a='BNT162b2SA (30 mcg, 1 Dose)' then
            do;
            actarmcd='B2_P23_30-SA_30_1DB';
            actarm='BNT162b2 Phase 2/3 (30 mcg) - BNT162b2SA (30 mcg) 1 booster';
         end;

      if trt01a='BNT162b2 Phase 2/3 (30 mcg)' and
         trt02a='BNT162b2SA (30 mcg, 2 Doses)' then
            do;
            actarmcd='B2_P23_30-SA_30_2DB';
            actarm='BNT162b2 Phase 2/3 (30 mcg) - BNT162b2SA (30 mcg) 2 booster';
         end;
```

```
        if trt01a='BNT162b2 Phase 2/3 (30 mcg)' and
           trt02a='BNT162b2 (5 mcg, 1 Dose)' then
              do;
              actarmcd='B2_P23_30-B2_5_1DB';
              actarm='BNT162b2 Phase 2/3 (30 mcg) - BNT162b2 (5 mcg) 1 booster';
           end;

        if trt01a='BNT162b2 Phase 2/3 (30 mcg)' and
           trt02a='BNT162b2 (10 mcg, 1 Dose)' then
              do;
              actarmcd='B2_P23_30-B2_10_1DB';
              actarm='BNT162b2 Phase 2/3 (30 mcg) - BNT162b2 (10 mcg) 1 booster';
           end;
     end;
  else if missing(trt01a) and not missing(trt02a) and index(arm, 'booster') then
     do;
        actarm='Unplanned Treatment';
        actarmcd='UNPLAN';
     end;
  *actarm UNPLAN for Indeterminate;

  if trt01a='Indeterminate' or (index(lowcase(arm), 'booster') and
     trt02a='Indeterminate')then
        do;
        actarm='Unplanned Treatment';
        actarmcd='UNPLAN';
     end;
  TRT02PN=input(TRT02P, trtfmt.);
  TRT02AN=input(TRT02A, trtfmt.);

  if randdt~=. then
     TRTSEQP=catx(' => ', trt01p, trt02p);

  if trtsdt~=. then
     do;
        TRTSEQA=catx(' => ', trt01a, trt02a);

        if missing(trt01a) and not missing(trt02a) then
           TRTSEQA=strip(trt02a);
     end;
run;

*******************************************************************************************;
* Specification 3 *;
* ADD PERIOD 2 DISPOSITION INFO *;
* 1 - UNBLNDDT/REVXICDT. *;
* 2 - EOT for Period 2. *;
*******************************************************************************************;
*Add UNBLNDDT from DSSTDTC where DSDECOD='TREATMENT UNBLINDED';

proc sql;
  create table _unblnd as select distinct usubjid, input(DSSTDTC, yymmdd10.) as
     UNBLNDDT format=date9. label="Treatment Unblinded Date" from DsSet where
```

```
      DSDECOD="TREATMENT UNBLINDED" and DSSTDTC ne "" order by usubjid, unblnddt;
quit;

data _unblnd;
   set _unblnd;
   by usubjid unblnddt;

   if first.usubjid then
      output;
run;

data adsl;
   merge adsl(in=a) _unblnd;
   by usubjid;

   if a;
run;

*Add revax icd;

data dsicd;
   set DsSet;
   where dsdecod="INFORMED CONSENT OBTAINED" and not missing(dsstdtc);
run;

proc sql undo_policy=none;
   create table dsicd as select *, count(usubjid) as icdcnt from dsicd group by
      usubjid order by usubjid, dsstdtc;
   create table dsicd as select a.*, b.unblnddt from dsicd a left join adsl b on
      a.usubjid=b.usubjid;
quit;

data dsicd;
   set dsicd;
   where not missing(unblnddt) and icdcnt>1;
run;

proc sort data=dsicd;
   by usubjid descending dsstdt;
run;

proc sort data=dsicd nodupkey;
   by usubjid;
run;

data adsl;
   merge adsl(in=a) dsicd(keep=usubjid dsstdt rename=(dsstdt=REVXICDT));
   label REVXICDT="Re-Informed Consent Date";
   by usubjid;

   if a;
run;

*Add EOT for open label;
```

```
proc sql;
   create table dsopen as select a.*, b.qval as dsphase from dataprot.ds a left
      join dataprot.suppds b on a.usubjid=b.usubjid and a.dsseq=input(b.idvarval,
      best.) order by usubjid;
quit;

data dsopen;
   set dsopen;
   Attrib EotXDcDt Label="End Of Open Label Treatment D/C Date"
      Format=date9. EotXDcRs Label="End Of Open Label Treatment D/C Reason";
   EotXDcDt=input(dsstdtc, yymmdd10.);
   EotXDcRs=dsdecod;
   where DSCAT="DISPOSITION EVENT" and DSPHASE="OPEN LABEL TREATMENT" and DSDECOD
      ne "COMPLETED";
run;

data adsl;
   merge adsl(in=a) dsopen(keep=usubjid EotXDcDt EotXDcRs);
   by usubjid;

   if a;
run;

*****************************************************************************************;
* Specification 4 *;
* ADD OTHER BASELINE INFO *;
* 1 - BMI category. *;
* 2 - Comorbodities flag. *;
* 3 - COVID baseline info. *;
*****************************************************************************************;
*Add baseline BMI category;

data bmi;
   set dataprot.vs;
   where (index(visit, "V1_DAY1") or index(visit, "V401_DAY1_VAX1")) and usubjid
      ne "" and vstestcd="BMI" and vsdy<=1;
run;

data adsl;
   merge adsl (in=a) bmi(keep=usubjid vsstresn);
   by usubjid;

   if a;
   label BMICAT="Baseline BMI Category" BMICATN="Baseline BMI Category (N)";
   length BMICAT $20;

   if vsstresn=. then
      BMICAT="Missing";

   if .<vsstresn<18.5 then
      BMICAT="Underweight";
   else if 18.5<=vsstresn<25 then
      BMICAT="Normal weight";
```

```
    else if 25<=vsstresn<30 then
       BMICAT="Overweight";
    else if 30<=vsstresn then
       BMICAT="Obese";

    if BMICAT="Underweight" then
       BMICATN=1;
    else if BMICAT="Normal weight" then
       BMICATN=2;
    else if BMICAT="Overweight" then
       BMICATN=3;
    else if BMICAT="Obese" then
       BMICATN=4;
    else if BMICAT="Missing" then
       BMICATN=5;
run;

proc import datafile="&expath./bmi-12-15-scale.xlsx" out=adobmi dbms=xlsx
       replace;
    getnames=no;
    datarow=2;
run;

proc sort data=adobmi;
    by A descending B;
run;

data adobmim;
    set adobmi(where=(A="Male"));
    B1=lag(B);

    if missing(B1) then
       B1=9999;
run;

data adobmif;
    set adobmi(where=(A="Female"));
    B1=lag(B);

    if missing(B1) then
       B1=9999;
run;

data adobmi;
    set adobmif adobmim;
run;

proc sql undo_policy=none;
    create table adsl as select a.*, b.c as obscut from adsl a left join adobmi b
       on b.B<=aaged/30.4375<b.B1 and 12<=floor((a.rficdt-a.brthdt)/365.25)<16
       and ((a.sex="M" and b.A="Male") or (a.sex="F" and b.A="Female")) order by
       a.usubjid;
quit;
```

```
data adsl(drop=vsstresn obscut);
   set adsl;
   label OBESEFL="Obese Flag for Adolescent";

   if .<vsstresn<obscut then
      OBESEFL="N";
   else if vsstresn>=obscut>. then
      OBESEFL="Y";
run;

*ADSL for Comorbodities - Xstart;

%macro read_cci;
   data filelst;
      retain filenum 0;
      rc=filename("dirpdf", "&expath");
      openfile=dopen("dirpdf");

      if openfile>0 then
         do;
            nummem=dnum(openfile);

            do ii=1 to nummem;
               name=dread(openfile, ii);

               if index(upcase(name), "REPORT-CCI") and index(upcase(name),"XLSX") then
                  do;
                     filenum+1;
                     output;
                  end;
            end;
         end;
      call symput('filetot', filenum);
   run;

   %do i=1 %to &filetot;

      proc sql;
         select name into: filename separated by "" from filelst where filenum=&i;
      quit;

      proc import datafile="&expath./&filename" out=fileout dbms=xlsx replace;
         datarow=17;
         getnames=yes;
      run;

      data fileout(drop=_A_);
         set fileout(rename=(a=_A_));
         length a $100.;
         a=_A_;
      run;

      %if &i=1 %then
         %do;
```

```
            data out;
               set fileout;
            run;

         %end;
      %else
         %do;

            data out;
               set out fileout;
            run;

         %end;
   %end;
%mend;

%read_cci;

proc sort data=out dupout=dupp nodupkey;
   by B;
run;

proc sql;
   create table cci as select a.*, b.C from dataprot.mh a left join out b on
      a.MHPTCD=input(b.B, best.);
quit;

proc sort data=cci nodupkey;
   by usubjid;
   where not missing(C);
run;

data adsl;
   merge adsl(in=a) cci(in=b keep=usubjid);
   by usubjid;

   if a;
   label COMBODFL="Flag for Comorbodities";

   if b then
      COMBODFL="Y";
   else
      COMBODFL="N";
run;

*Xend;
*Get baseline information for efficacy;
*Deal with retest;

proc sql;
   create table is as select a.*, b.phasen from dataprot.is a left join adsl b on
      a.usubjid=b.usubjid;
quit;
```

```
* version 3, according to team, no need to remove the original result when derive evaluable flags, 12May2021;

proc sort data=is out=is_rep;
   by usubjid istestcd visitnum visit isdtc;
run;

data is;
   set is_rep;
   where (index(visit, "V1_DAY1_") or index(visit, "V401_DAY1_VAX1")) and
      istestcd in ('C19NIG');
   ;
run;

proc sort data=is;
   by usubjid descending isorres;
run;

proc sort data=is nodupkey;
   by usubjid;
run;

data isn;
   set is;
   where isorres="NEG";
run;

data adsl;
   merge adsl(in=a) is(in=b keep=usubjid isorres isdy);
   by usubjid;
   label NIGV1FL="N-binding Antibody Neg at Visit 1 Flag";

   if a;

   if isorres="NEG" and isdy<=1 then
      NIGV1FL="Y";
   else if isorres="POS" and isdy<=1 then
      NIGV1FL="N";
   drop isorres isdy;
run;

data mb;
   set dataprot.mb;
   where (index(visit, "V1_DAY1_") or index(visit, "V401_DAY1_VAX1")) and
      mbtestcd='RTCOV2NS';
run;

proc sort data=mb nodupkey;
   by usubjid;
run;

data mbn;
   set mb;
   where mborres="NEG";
```

```
run;

data adsl;
   merge adsl(in=a) mb(in=b keep=usubjid mborres mbdy);
   by usubjid;
   label NAATNFL="NAAT Negative at Visit 1 Flag";

   if a;

   if mborres="NEG" and mbdy<=1 then
      NAATNFL="Y";
   else if mborres="POS" and mbdy<=1 then
      NAATNFL="N";
   drop mborres mbdy;
run;

data mh;
   set dataprot.mh;
   where MHDECOD in ("Asymptomatic COVID-19" "COVID-19" "COVID-19 pneumonia"
      "COVID-19 treatment" "Suspected COVID-19" "SARS-CoV-2 antibody test positive"
      "SARS-CoV-2 carrier" "SARS-CoV-2 sepsis" "SARS-CoV-2 test positive"
      "SARS-CoV-2 viraemia" "Multisystem inflammatory syndrome in children");
run;

data adsl(drop=isorres mborres isdy mbdy);
   merge adsl(in=a) mh(in=b keep=usubjid) mb(in=c keep=usubjid mborres mbdy)
      is(in=d keep=usubjid isorres isdy);
   by usubjid;

   if a;
   label COVBLST="Baseline SARS-CoV-2 Status";

   if b or (mborres="POS" and mbdy<=1) or (isorres="POS" and isdy<=1) then
      COVBLST="POS";

   if not b and (mborres="NEG" and mbdy<=1) and (isorres="NEG" and isdy<=1) then
      COVBLST="NEG";
run;

*******************************************************************************;
* Specification 5 *;
* IMMUNOGENICITY DATA *;
* 1 - Figure out subjects without V3/V4 but with COVID in window. *;
* 2 - Replace V3/V4 using convalescent visits in window. *;
* 3 - Combine info from CO and iS. *;
*******************************************************************************;
*Subjects with convalescent visits meeting requirement;

data co_rep co_other;
   set dataprot.co;

   if rdomain='IS' and coval='N' and coref='Sample Collected' then
      output co_rep;
   else
```

```
    output co_other;
run;

data cocovall;
   length covdtc $20.;
   set dataprot.co;
   where (index(visit, "COVID") or upcase(strip(visit))='V101_VAX3' or
      upcase(strip(visit))='V201_SURVEIL_CONSENT') and rdomain="IS" and
      COREF='Sample Collected' and COVAL="Y";
   covvis=visit;
   covdtc=codtc;
   covdt=input(codtc, e8601da.);
   covvisnum=visitnum;
   format covdt is8601da.;
run;

data pend_rep;
   set co_rep;
   where visit="V302_WEEK1_POSTVAX3";
run;

proc sql;
   create table merge_sv as select a.*, b.svstdtc, c.vax201dt from pend_rep a
      left join (select distinct usubjid, visit, svstdtc from dataprot.sv)b on
      a.usubjid=b.usubjid and a.visit=b.visit left join adsl c on
      a.usubjid=c.usubjid;
quit;

data rep_1;
   set merge_sv;
   obs=_n_;

   if not missing(svstdtc) then
      do;
         sv_target=input(svstdtc, e8601da.);
         sv_lo=sv_target - 7;
         sv_hi=sv_target + 7;
      end;

   if not missing(vax201dt) then
      do;
         dos_lo=vax201dt + 6;
         dos_hi=vax201dt + 8;
         dos_target=vax201dt + 7;
      end;
   format sv_target sv_lo sv_hi dos_lo dos_hi dos_target is8601da.;
run;

proc sql;
   create table rep_2 as select a.*, b.covdt, b.covvis, b.covvisnum, b.covdtc
      from rep_1 a left join cocovall b on a.usubjid=b.usubjid
      and ((.< a.sv_lo <=b.covdt <=a.sv_hi) or (.< a.dos_lo <=b.covdt <=a.dos_hi))
      order by obs;
quit;
```

```
data rep_3;
   set rep_2;

   if not missing(covdt) then
      do;

         if not missing(sv_target) then
            diff=abs(covdt - sv_target);
         else if not missing(dos_target) then
            diff=abs(covdt - dos_target);
         else
            diff=99999;
      end;

proc sort;
   by obs diff;
run;

data rep_a;
   set rep_3;
   by obs diff;

   if not missing(covdt) then
      do;
         codtc=put(covdt, is8601da.);
         coval='Y';
         replace_fl='Y';
      end;

   if first.obs then
      output;
run;

;

data pend_rep;
   set co_rep;
   where visit="V303_MONTH1_POSTVAX3";
run;

proc sql;
   create table merge_sv as select a.*, b.svstdtc, c.vax201dt from pend_rep a
      left join (select distinct usubjid, visit, svstdtc from dataprot.sv)b on
      a.usubjid=b.usubjid and a.visit=b.visit left join adsl c on
      a.usubjid=c.usubjid;
quit;

data rep_1;
   set merge_sv;
   obs=_n_;

   if not missing(svstdtc) then
      do;
```

```
        sv_target=input(svstdtc, e8601da.);
        sv_lo=sv_target - 7;
        sv_hi=sv_target + 7;
      end;

   if not missing(vax201dt) then
      do;
        dos_lo=vax201dt + 28;
        dos_hi=vax201dt + 42;
        dos_target=vax201dt + 30;
      end;
   format sv_target sv_lo sv_hi dos_lo dos_hi dos_target is8601da.;
run;

proc sql;
   create table rep_2 as select a.*, b.covdt, b.covvis, b.covvisnum, b.covdtc
      from rep_1 a left join cocovall b on a.usubjid=b.usubjid
      and ((.< a.sv_lo <=b.covdt <=a.sv_hi) or (.< a.dos_lo <=b.covdt <=a.dos_hi))
      order by obs;
quit;

data rep_3;
   set rep_2;

   if not missing(covdt) then
      do;

        if not missing(sv_target) then
          diff=abs(covdt - sv_target);
        else if not missing(dos_target) then
          diff=abs(covdt - dos_target);
        else
          diff=99999;
      end;

proc sort;
   by obs diff;
run;

data rep_b;
   set rep_3;
   by obs diff;

   if not missing(covdt) then
      do;
        codtc=put(covdt, is8601da.);
        coval='Y';
        replace_fl='Y';
      end;

   if first.obs then
      output;
run;
```

```
;

data pend_rep;
   set co_rep;
   where visit="V3_MONTH1_POSTVAX2_L";
run;

proc sql;
   create table merge_sv as select a.*, b.svstdtc, c.vax102dt from pend_rep a
      left join (select distinct usubjid, visit, svstdtc from dataprot.sv)b on
      a.usubjid=b.usubjid and a.visit=b.visit left join adsl c on
      a.usubjid=c.usubjid;
quit;

data rep_1;
   set merge_sv;
   obs=_n_;

   if not missing(svstdtc) then
      do;
         sv_target=input(svstdtc, e8601da.);
         sv_lo=sv_target - 7;
         sv_hi=sv_target + 7;
      end;

   if not missing(vax102dt) then
      do;
         dos_lo=vax102dt + 28;
         dos_hi=vax102dt + 42;
         dos_target=vax102dt + 30;
      end;
   format sv_target sv_lo sv_hi dos_lo dos_hi dos_target is8601da.;
run;

proc sql;
   create table rep_2 as select a.*, b.covdt, b.covvis, b.covvisnum, b.covdtc
      from rep_1 a left join cocovall b on a.usubjid=b.usubjid
      and ((.< a.sv_lo <=b.covdt <=a.sv_hi) or (.< a.dos_lo <=b.covdt <=a.dos_hi))
      order by obs;
quit;

data rep_3;
   set rep_2;

   if not missing(covdt) then
      do;

         if not missing(sv_target) then
            diff=abs(covdt - sv_target);
         else if not missing(dos_target) then
            diff=abs(covdt - dos_target);
         else
            diff=99999;
      end;
```

```
proc sort;
   by obs diff;
run;

data rep_c;
   set rep_3;
   by obs diff;

   if not missing(covdt) then
      do;
         codtc=put(covdt, is8601da.);
         coval='Y';
         replace_fl='Y';
      end;

   if first.obs then
      output;
run;

;

data pend_rep;
   set co_rep;
   where visit="V4_MONTH6_L";
run;

proc sql;
   create table merge_sv as select a.*, b.svstdtc, c.vax102dt from pend_rep a
      left join (select distinct usubjid, visit, svstdtc from dataprot.sv)b on
      a.usubjid=b.usubjid and a.visit=b.visit left join adsl c on
      a.usubjid=c.usubjid;
quit;

data rep_1;
   set merge_sv;
   obs=_n_;

   if not missing(svstdtc) then
      do;
         sv_target=input(svstdtc, e8601da.);
         sv_lo=sv_target - 7;
         sv_hi=sv_target + 7;
      end;

   if not missing(vax102dt) then
      do;
         dos_lo=vax102dt + 175;
         dos_hi=vax102dt + 189;
         dos_target=vax102dt + 180;
      end;
   format sv_target sv_lo sv_hi dos_lo dos_hi dos_target is8601da.;
run;
```

```
proc sql;
   create table rep_2 as select a.*, b.covdt, b.covvis, b.covvisnum, b.covdtc
      from rep_1 a left join cocovall b on a.usubjid=b.usubjid
      and ((.< a.sv_lo <=b.covdt <=a.sv_hi) or (.< a.dos_lo <=b.covdt <=a.dos_hi))
      order by obs;
quit;

data rep_3;
   set rep_2;

   if not missing(covdt) then
      do;

         if not missing(sv_target) then
            diff=abs(covdt - sv_target);
         else if not missing(dos_target) then
            diff=abs(covdt - dos_target);
         else
            diff=99999;
      end;

proc sort;
   by obs diff;
run;

data rep_d;
   set rep_3;
   by obs diff;

   if not missing(covdt) then
      do;
         codtc=put(covdt, is8601da.);
         coval='Y';
         replace_fl='Y';
      end;

   if first.obs then
      output;
run;

;

data pend_rep;
   set co_rep;
   where visit="V7_MONTH1_S";
run;

proc sql;
   create table merge_sv as select a.*, b.svstdtc, c.vax102dt from pend_rep a
      left join (select distinct usubjid, visit, svstdtc from dataprot.sv)b on
      a.usubjid=b.usubjid and a.visit=b.visit left join adsl c on
      a.usubjid=c.usubjid;
quit;
```

```
data rep_1;
   set merge_sv;
   obs=_n_;

   if not missing(svstdtc) then
      do;
         sv_target=input(svstdtc, e8601da.);
         sv_lo=sv_target - 7;
         sv_hi=sv_target + 7;
      end;

   if not missing(vax102dt) then
      do;
         dos_lo=vax102dt + 28;
         dos_hi=vax102dt + 42;
         dos_target=vax102dt + 30;
      end;
   format sv_target sv_lo sv_hi dos_lo dos_hi dos_target is8601da.;
run;

proc sql;
   create table rep_2 as select a.*, b.covdt, b.covvis, b.covvisnum, b.covdtc
      from rep_1 a left join cocovall b on a.usubjid=b.usubjid
      and ((.< a.sv_lo <=b.covdt <=a.sv_hi) or (.< a.dos_lo <=b.covdt <=a.dos_hi))
      order by obs;
quit;

data rep_3;
   set rep_2;

   if not missing(covdt) then
      do;

         if not missing(sv_target) then
            diff=abs(covdt - sv_target);
         else if not missing(dos_target) then
            diff=abs(covdt - dos_target);
         else
            diff=99999;
      end;

proc sort;
   by obs diff;
run;

data rep_e;
   set rep_3;
   by obs diff;

   if not missing(covdt) then
      do;
         codtc=put(covdt, is8601da.);
         coval='Y';
         replace_fl='Y';
```

```
        end;

    if first.obs then
        output;
run;

;

data pend_rep;
    set co_rep;
    where visit="V8_MONTH6_S";
run;

proc sql;
    create table merge_sv as select a.*, b.svstdtc, c.vax102dt from pend_rep a
        left join (select distinct usubjid, visit, svstdtc from dataprot.sv)b on
        a.usubjid=b.usubjid and a.visit=b.visit left join adsl c on
        a.usubjid=c.usubjid;
quit;

data rep_1;
    set merge_sv;
    obs=_n_;

    if not missing(svstdtc) then
        do;
            sv_target=input(svstdtc, e8601da.);
            sv_lo=sv_target - 7;
            sv_hi=sv_target + 7;
        end;

    if not missing(vax102dt) then
        do;
            dos_lo=vax102dt + 175;
            dos_hi=vax102dt + 189;
            dos_target=vax102dt + 180;
        end;
    format sv_target sv_lo sv_hi dos_lo dos_hi dos_target is8601da.;
run;

proc sql;
    create table rep_2 as select a.*, b.covdt, b.covvis, b.covvisnum, b.covdtc
        from rep_1 a left join cocovall b on a.usubjid=b.usubjid
        and ((.< a.sv_lo <=b.covdt <=a.sv_hi) or (.< a.dos_lo <=b.covdt <=a.dos_hi))
        order by obs;
quit;

data rep_3;
    set rep_2;

    if not missing(covdt) then
        do;

            if not missing(sv_target) then
```

```
            diff=abs(covdt - sv_target);
         else if not missing(dos_target) then
            diff=abs(covdt - dos_target);
         else
            diff=99999;
      end;

proc sort;
   by obs diff;
run;

data rep_f;
   set rep_3;
   by obs diff;

   if not missing(covdt) then
      do;
         codtc=put(covdt, is8601da.);
         coval='Y';
         replace_fl='Y';
      end;

   if first.obs then
      output;
run;

;

data pend_rep;
   set co_rep;
   where visit="V404_MONTH1_POSTVAX2";
run;

proc sql;
   create table merge_sv as select a.*, b.svstdtc, c.vax102dt from pend_rep a
      left join (select distinct usubjid, visit, svstdtc from dataprot.sv)b on
      a.usubjid=b.usubjid and a.visit=b.visit left join adsl c on
      a.usubjid=c.usubjid;
quit;

data rep_1;
   set merge_sv;
   obs=_n_;

   if not missing(svstdtc) then
      do;
         sv_target=input(svstdtc, e8601da.);
         sv_lo=sv_target - 7;
         sv_hi=sv_target + 7;
      end;

   if not missing(vax102dt) then
      do;
         dos_lo=vax102dt + 28;
```

```
        dos_hi=vax102dt + 42;
        dos_target=vax102dt + 30;
      end;
   format sv_target sv_lo sv_hi dos_lo dos_hi dos_target is8601da.;
run;

proc sql;
   create table rep_2 as select a.*, b.covdt, b.covvis, b.covvisnum, b.covdtc
      from rep_1 a left join cocovall b on a.usubjid=b.usubjid
      and ((.< a.sv_lo <=b.covdt <=a.sv_hi) or (.< a.dos_lo <=b.covdt <=a.dos_hi))
      order by obs;
quit;

data rep_3;
   set rep_2;

   if not missing(covdt) then
      do;

         if not missing(sv_target) then
            diff=abs(covdt - sv_target);
         else if not missing(dos_target) then
            diff=abs(covdt - dos_target);
         else
            diff=99999;
      end;

proc sort;
   by obs diff;
run;

data rep_g;
   set rep_3;
   by obs diff;

   if not missing(covdt) then
      do;
         codtc=put(covdt, is8601da.);
         coval='Y';
         replace_fl='Y';
      end;

   if first.obs then
      output;
run;

;

data co_replace;
   set co_other co_rep(where=(visit not in ('V302_WEEK1_POSTVAX3'
      'V303_MONTH1_POSTVAX3' 'V3_MONTH1_POSTVAX2_L' 'V4_MONTH6_L' 'V7_MONTH1_S'
      'V8_MONTH6_S' 'V404_MONTH1_POSTVAX2'))) rep_a rep_b rep_c rep_d rep_e rep_f
      rep_g;
```

```
    if not missing(codtc) then
       codate=input(codtc, e8601da.);
    format codate is8601da.;
run;

*replace v301 by v4;

data v301_miss v301_ex;
    set co_replace;

    if coval='N' and rdomain='IS' and coref='Sample Collected' and
       visit='V301_VAX3' then
          output v301_miss;
    else
       output v301_ex;
run;

proc sql;
    create table v301_miss2 as select a.*, b.codtc as v4dtc, c.vax201dt as dose3dt
       from v301_miss a left join (select usubjid, codtc from dataprot.co where
       coval='Y' and rdomain='IS' and coref="Sample Collected" and
       visit='V4_MONTH6_L') b on a.usubjid=b.usubjid left join adsl c on
       a.usubjid=c.usubjid;
quit;

data v301_miss3;
    set v301_miss2;

    if not missing(v4dtc) and not missing(dose3dt) then
       do;
          codtc=v4dtc;
          coval='Y';
       end;
run;

data co_replace;
    set v301_miss3 v301_ex;

    if not missing(codtc) then
       codate=input(codtc, e8601da.);
    format codate is8601da.;

    if rdomain='IS' and coval='Y' and coref='Sample Collected';
run;

data co_rep_record;
    set co_replace;

    if not missing(covdtc);
run;

proc sql;
    create table isnew0 as select a.*, b.visit as mvis, b.visitnum as mvisnum,
       b.covvisnum from is_rep a left join co_rep_record b on a.usubjid=b.usubjid
```

```
      and a.visitnum=b.covvisnum and a.isdtc=b.covdtc;
quit;

data isnew;
   set isnew0;

   if not missing(covvisnum) and (index(visit, "COVID") or
      upcase(strip(visit))='V101_VAX3' or
      upcase(strip(visit))='V201_SURVEIL_CONSENT') then
         do;
         visit=mvis;
         visitnum=mvisnum;
      end;
run;

proc sql;
   create table isva as select a.*, b.cohortn, b.vax101dt, b.vax102dt,
      b.vax10Udt, b.vax201dt , b.vax202dt from isnew a left join adsl b on
      a.usubjid=b.usubjid order by usubjid;
quit;

data isva;
   set isva;
   format isdate date9.;

   if not missing(isdtc) then
      isdate=input(isdtc, yymmdd10.);

   if not missing(vax202dt) and index(visit, 'POSTVAX4') then
      do;
         novax=4;
         diff=isdate-vax202dt;
      end;
   else if not missing(vax201dt) and index(visit, 'POSTVAX3') then
      do;
         novax=3;
         diff=isdate-vax201dt;
      end;
   else if not missing(vax102dt) and index(visit, "_VAX1")=0 and index(visit,
      "_POSTVAX1")=0 and index(visit, "_VAX2")=0 then
         do;
         novax=2;
         diff=isdate-vax102dt;
      end;
   else if not missing(vax101dt) then
      do;
         novax=1;
         diff=isdate-vax101dt;
      end;
run;

proc sort data=co_replace out=co_replace_1(keep=usubjid coval visitnum visit)
      nodupkey;
   by usubjid visitnum;
```

```
run;

proc transpose data=co_replace_1 out=t_co_replace_1 prefix=covis;
   by usubjid;
   id visitnum;
   idlabel visit;
   var coval;
run;

proc sort data=co_replace out=co_replace_2(keep=usubjid codate visitnum visit)
      nodupkey;
   by usubjid visitnum;
run;

proc transpose data=co_replace_2 out=t_co_replace_2 prefix=codt;
   by usubjid;
   id visitnum;
   idlabel visit;
   var codate;
run;

*******************************************************************************************;
* Specification 6 *;
* ADVERSE EVENT CUTOFF *;
* 1 -1/6 month(s) follow up date in double blind period from blood sample. *;
* 2 -1/6 month(s) follow up date in open label period from blood sample. *;
* 3 -1/6 month(s) follow up date in double blind period from SV. *;
* 4 -1/6 month(s) follow up date in open label period from SV. *;
*******************************************************************************************;

proc sql;
   create table sv as select a.*, input(svstdtc, yymmdd10.) as svstdt, b.cohortn
      from dataprot.sv a left join adsl b on a.usubjid=b.usubjid;
quit;

proc sql;
   create table co_m as select a.*, b.cohortn from dataprot.co a left join adsl b
      on a.usubjid=b.usubjid;
quit;

data v0xdt;
   set co_m;

   if coref='Sample Collected';

   if not missing(codtc);
   bedt=input(codtc, yymmdd10.);
   where not (cohortn=1.16 and visitnum in (60751 60752 60753 60754));

proc sort nodupkey;
   by usubjid visitnum codtc;
run;

data adsl;
```

```
merge adsl (in=a) v0xdt(keep=usubjid bedt visit cohortn rename=(bedt=be1dt)
   where=(index(visit, "V103_")=0 and ((index(visit, "_MONTH1_") and cohortn ne
   1.16 and index(visit, 'POSTVAX3') eq 0 and index(visit, 'POSTVAX4') eq 0)
   or (visit="V7_MONTH1_S_R") and cohortn=1.16))) v0xdt(keep=usubjid bedt visit
   cohortn rename=(bedt=be2dt) where=(index(visit, "V104_")=0 and index(visit,
   "_MONTH6_"))) v0xdt(keep=usubjid bedt visit cohortn rename=(bedt=be3dt)
   where=(visit in ("V103_MONTH1" "V305_MONTH1_POSTVAX4"))) v0xdt(keep=usubjid
   bedt visit cohortn rename=(bedt=be3_1dt) where=(visit
   in ("V8C_MONTH1_POSTVAX3" "V303_MONTH1_POSTVAX3"))) v0xdt(keep=usubjid bedt
   visit cohortn rename=(bedt=be4dt) where=(visit in ("V104_MONTH6")))
   v0xdt(keep=usubjid bedt visit cohortn rename=(bedt=be4_1dt) where=(visit
   in ("V306_MONTH6"))) sv(keep=usubjid svstdt visit cohortn
   rename=(svstdt=be1dt2) where=(index(visit, "V103_")=0 and ((index(visit,
   "_MONTH1_") and cohortn ne 1.16 and index(visit, 'POSTVAX3') eq 0 and
   index(visit, 'POSTVAX4') eq 0) or (visit="V7_MONTH1_S_R") and cohortn=1.16)))
   sv(keep=usubjid svstdt visit cohortn rename=(svstdt=be2dt2)
   where=(index(visit, "V104_")=0 and index(visit, "_MONTH6_"))) sv(keep=usubjid
   svstdt visit cohortn rename=(svstdt=be3dt2) where=(visit in ("V103_MONTH1"
   "V305_MONTH1_POSTVAX4"))) sv(keep=usubjid svstdt visit cohortn
   rename=(svstdt=be3_1dt2) where=(visit in ("V8C_MONTH1_POSTVAX3"
   "V303_MONTH1_POSTVAX3"))) sv(keep=usubjid svstdt visit cohortn
   rename=(svstdt=be4dt2) where=(visit in ("V104_MONTH6"))) sv(keep=usubjid
   svstdt visit cohortn rename=(svstdt=be4_1dt2) where=(visit
   in ("V306_MONTH6")));
by usubjid;

if a;
attrib V01DT label="Date of Unblinding or Visit at 1MPD2" format=date9. V02DT
   label="Date of Unblinding or Visit at 6MPD2" format=date9. V02OBDT
   label="Date of Dose 3 or Visit at 6MPD2" format=date9. V03DT
   label="Date of Visit at 1M after Vax3/4" format=date9. V04DT
   label="Date of Visit at 6M after Vax3/4" format=date9.;

if not missing(VAX10UDT) and VAX10UDT>VAX102DT then
   V01DT=VAX10UDT+35;
else if not missing(be1dt) then
   V01DT=be1dt;
else if not missing(be1dt2) then
   V01DT=be1dt2;
else if not missing(VAX102DT) then
   V01DT=VAX102DT+35;
else if not missing(VAX101DT) then
   V01DT=VAX101DT+58;

if not missing(VAX10UDT) and VAX10UDT>VAX102DT then
   V02DT=VAX10UDT+189;
else if not missing(be2dt) then
   V02DT=be2dt;
else if not missing(be2dt2) then
   V02DT=be2dt2;
else if not missing(VAX102DT) then
   V02DT=VAX102DT+189;
else if not missing(VAX101DT) then
   V02DT=VAX101DT+189+23;
```

```
*Cutoff V01DT V02DT by UNBLNDDT Treatment unblinded Date;

if (v02dt>=tr02sdt>. or (v02dt=. and tr02sdt>.)) then
   V02OBDT=tr02sdt-1;
else
   V02OBDT=V02DT;

if UNBLNDDT~=. then
   do;

      if V01DT~=. and V01DT>(UNBLNDDT-1) then
         V01DT=UNBLNDDT-1;

      if V02DT~=. and V02DT>(UNBLNDDT-1) then
         V02DT=UNBLNDDT-1;
   end;

if index(trt02p, '1 Dose') then
   do;

      if not missing(VAX20UDT) and VAX20UDT>VAX201DT then
         V03DT=VAX20UDT+35;
      else if not missing(be3_1dt) then
         V03DT=be3_1dt;
      else if not missing(be3_1dt2) then
         V03DT=be3_1dt2;
      else if not missing(VAX201DT) then
         V03DT=VAX201DT+35;

      if not missing(VAX20UDT) and VAX20UDT>VAX201DT then
         V04DT=VAX20UDT+189;
      else if not missing(be4_1dt) then
         V04DT=be4_1dt;
      else if not missing(be4_1dt2) then
         V04DT=be4_1dt2;
      else if not missing(VAX201DT) then
         V04DT=VAX201DT+189;
   end;
else if index(trt02p, '2 Doses') then
   do;

      if not missing(VAX20UDT) and VAX20UDT>VAX202DT then
         V03DT=VAX20UDT+35;
      else if not missing(be3dt) then
         V03DT=be3dt;
      else if not missing(be3dt2) then
         V03DT=be3dt2;
      else if not missing(VAX202DT) then
         V03DT=VAX202DT+35;
      else if not missing(VAX201DT) then
         V03DT=VAX201DT+35+35;

      if not missing(VAX20UDT) and VAX20UDT>VAX202DT then
         V04DT=VAX20UDT+189;
```

```
        else if not missing(be4dt) then
           V04DT=be4dt;
        else if not missing(be4dt2) then
           V04DT=be4dt2;
        else if not missing(VAX202DT) then
           V04DT=VAX202DT+189;
        else if not missing(VAX201DT) then
           V04DT=VAX201DT+189+35;
     end;
  else
     do;

        if not missing(VAX20UDT) and VAX20UDT>VAX202DT then
           V03DT=VAX20UDT+35;
        else if not missing(be3dt) then
           V03DT=be3dt;
        else if not missing(be3dt2) then
           V03DT=be3dt2;
        else if not missing(VAX202DT) then
           V03DT=VAX202DT+35;
        else if not missing(VAX201DT) then
           V03DT=VAX201DT+58;

        if not missing(VAX20UDT) and VAX20UDT>VAX202DT then
           V04DT=VAX20UDT+189;
        else if not missing(be4dt) then
           V04DT=be4dt;
        else if not missing(be4dt2) then
           V04DT=be4dt2;
        else if not missing(VAX202DT) then
           V04DT=VAX202DT+189;
        else if not missing(VAX201DT) then
           V04DT=VAX201DT+189+23;
     end;

  if missing(vax201dt) and missing(vax202dt) and missing(vax20udt) then
     do;
        V03DT=.;
        V04DT=.;
     end;
run;

proc sort data=adsl;
   by usubjid;
run;

*****************************************************************************************;
* Specification 7 *;
* BLOOD SAMPLE DRAWN *;
* 1 - Blood sampel drawn date and flags. *;
* 2 - Inclusion/exclusion flags. *;
* 3 - PD & Immuno pop flags. *;
*****************************************************************************************;
*Blood sample obtained from CO;
```

```
proc sort data=isva out=isva1(keep=usubjid diff visitnum visit) nodupkey;
   by usubjid visitnum;
   where isorres not in ("" "NOT DONE") and not missing(istest);
run;

proc transpose data=isva1 out=t_isva1 prefix=vis;
   by usubjid;
   id visitnum;
   idlabel visit;
   var diff;
run;

proc sort data=isva out=isva2(keep=usubjid isdate visitnum visit) nodupkey;
   by usubjid visitnum;
   where isorres not in ("" "NOT DONE") and not missing(istest);
run;

proc transpose data=isva2 out=t_isva2 prefix=isdt;
   by usubjid;
   id visitnum;
   idlabel visit;
   var isdate;
run;

proc sql;
   create table dv as select a.usubjid, a.dvstdtc, a.dvseq, a.epoch as epoch_dv,
      b.qval as cape from dataprot.dv a right join
      dataprot.suppdv(where=(QNAM="CAPE" and upcase(QVAL) not in (" " "NO"))) b on
      a.usubjid=b.usubjid and a.dvseq=input(b.idvarval, best.) order by usubjid,
      dvseq;
quit;

proc sort data=adsl;
   by usubjid;
run;

data dv;
   merge dv(in=a) adsl(in=b keep=usubjid vax201dt);
   by usubjid;

   if a;

   if length(dvstdtc) ge 10 then
      _dvdt=input(dvstdtc, e8601da.);
run;

proc sql;
   create table dvdate as select distinct usubjid, min(input(dvstdtc, yymmdd10.))
      as dvstdt label="Start Date of Important PD" format=date9. from dv where
      dvstdtc ne "" group by usubjid;
   create table dvdate_p1 as select distinct usubjid, min(input(dvstdtc,
      yymmdd10.)) as POP1PDDT label="First PD date for safety" format=date9. from
      dv where dvstdtc ne "" and index(cape, "POP1") group by usubjid;
```

```
    create table dvdate_p2 as select distinct usubjid, min(input(dvstdtc,
       yymmdd10.)) as POP2PDDT label="First PD date for efficacy" format=date9. from
       dv where dvstdtc ne "" and index(cape, "POP2") group by usubjid;
    create table dvdate_p3 as select distinct usubjid, min(input(dvstdtc,
       yymmdd10.)) as POP3PDDT label="First PD date for immunogenicity"
       format=date9. from dv where dvstdtc ne "" and index(cape, "POP3") group by
       usubjid;
    create table period1_pop3 as select distinct usubjid from dv where dvstdtc ne
       "" and index(cape, "POP3") and ((.<_dvdt < vax201dt) or missing(vax201dt))
       order by usubjid;
    create table dvout_0 as select distinct a.usubjid, b.dvstdt, c.usubjid as
       safety, d.usubjid as efficacy, e.usubjid as immuno, f.usubjid as multiple,
       g.usubjid as siteexcld from dv a left join dvdate b on a.usubjid=b.usubjid
       left join (select distinct usubjid from dv where index(cape, "POP1")) c on
       a.usubjid=c.usubjid left join (select distinct usubjid from dv where
       index(cape, "POP2")) d on a.usubjid=d.usubjid left join (select distinct
       usubjid from dv where index(cape, "POP3")) e on a.usubjid=e.usubjid left
       join (select distinct usubjid from dv where index(cape, "POP4")) f on
       a.usubjid=f.usubjid left join (select distinct usubjid from dv where
       index(cape, "POP5")) g on a.usubjid=g.usubjid;
    create table dvout as select a.*, b.pop1pddt, c.pop2pddt, d.pop3pddt,
       e.usubjid as period1_pop3 from dvout_0 a left join dvdate_p1 b on
       a.usubjid=b.usubjid left join dvdate_p2 c on a.usubjid=c.usubjid left join
       dvdate_p3 d on a.usubjid=d.usubjid left join period1_pop3 e on
       a.usubjid=e.usubjid;
quit;

proc sort data=dataprot.ie out=ie(keep=usubjid domain) nodupkey;
   by usubjid;
   where visit not in ("V101_VAX3" "V301_VAX3");
run;

*have N-binding antibody test result available at the 1-month post-Dose 2 visit;

data nbind;
   set isva;
   label V3C19NIG="C19NIG Result at Visit 3";
   V3C19NIG=isorres;
   keep usubjid V3C19NIG;
   where not missing(isorres) and istestcd="C19NIG" and index(visit, "V103_")=0
      and ((index(visit, "_MONTH1_") and index(visit, 'POSTVAX3')=0 and
      index(visit, 'POSTVAX4')=0 and cohortn ne 1.16) or (visit="V7_MONTH1_S_R")
      and cohortn=1.16);
run;

proc sort data=nbind nodupkey;
   by usubjid;
run;

*Valid IS result after Dose 1 but before Dose 2/after Dose 2 - Planned visits;

data incl3p;
   set isva;
```

```
    if istestcd^="C19NIG" and isorres not in ("" "IND" "QNS" "NOT DONE");

    if not missing(vax101dt) and isdate > vax101dt;

    if isdate <=vax102dt or (vax102dt=. and vax201dt ne . and vax101dt <=vax201dt)
       or (vax102dt=. and vax201dt=. and vax202dt ne . and vax101dt <=vax202dt)
       or (vax102dt=. and vax201dt=. and vax202dt=.);

proc sort nodupkey;
   by usubjid;
run;

data incl8p;
   set isva;

   if istestcd^="C19NIG" and isorres not in ("" "IND" "QNS" "NOT DONE");

   if not missing(vax102dt) and isdate > vax102dt;

   if isdate <=vax201dt or (vax201dt=. and vax202dt ne . and isdate <=vax202dt)
      or (vax201dt=. and vax202dt=.);

proc sort nodupkey;
   by usubjid;
run;

*Valid IS result after Dose 1 but before Dose 2/after Dose 2 - All;

proc sql;
   create table adsl_tmp as select a.*, b.svstdtc as v5p1, c.svstdtc as v3p2 from
      adsl a left join dataprot.sv b on a.usubjid=b.usubjid and a.phasen=1
      and ((b.visit='V5_WEEK1_POSTVAX2_S' and a.cohortn ne 1.16)
      or (b.visit='V5_WEEK1_POSTVAX2_S_R' and a.cohortn eq 1.16)) left join
      dataprot.sv c on a.usubjid=c.usubjid and phasen ne 1
      and (c.visit='V3_MONTH1_POSTVAX2_L' or c.visit='V404_MONTH1_POSTVAX2') order
      by a.usubjid;
quit;

data adsl;
   set adsl_tmp;

proc sort;
   by usubjid;
run;

*RSEXSAF;

data pop5_0;
   set dv;

   if index(cape, 'YES-POP5');

proc sort;
   by usubjid;
```

```
run;

data pop5_1;
   set pop5_0;
   by usubjid;
   length comb $200.;
   retain comb;

   if first.usubjid then
      comb=cape;
   else
      comb=catx(';', comb, cape);

   if last.usubjid then
      output;
run;

data pop5_2(keep=usubjid pop5_txt);
   length pop5_txt $200.;
   set pop5_1;

   if index(comb, 'YES-POP1') and index(comb, 'YES-POP2') and index(comb,
      'YES-POP3') then
         pop5_txt='Unreliable data due to lack of PI oversight - Safety, Efficacy, Immunogenicity Population(s)';
   else if index(comb, 'YES-POP1') and index(comb, 'YES-POP2') then
      pop5_txt='Unreliable data due to lack of PI oversight - Safety, Efficacy Population(s)';
   else if index(comb, 'YES-POP1') and index(comb, 'YES-POP3') then
      pop5_txt='Unreliable data due to lack of PI oversight - Safety, Immunogenicity Population(s)';
   else if index(comb, 'YES-POP2') and index(comb, 'YES-POP3') then
      pop5_txt='Unreliable data due to lack of PI oversight - Efficacy, Immunogenicity Population(s)';
   else if index(comb, 'YES-POP1') then
      pop5_txt='Unreliable data due to lack of PI oversight - Safety Population(s)';
   else if index(comb, 'YES-POP2') then
      pop5_txt='Unreliable data due to lack of PI oversight - Efficacy Population(s)';
   else if index(comb, 'YES-POP3') then
      pop5_txt='Unreliable data due to lack of PI oversight - Immunogenicity Population(s)';
   else
      pop5_txt='';

proc sort;
   by usubjid;
run;

data adsl(drop=pop5_txt);
   merge adsl(in=a) ie(rename=(domain=INEX)) t_isva1 t_isva2 t_co_replace_1
      t_co_replace_2 nbind sv(keep=usubjid visit svstdt cohortn
      rename=(svstdt=visit3dt) where=(index(visit, "V103_")=0 and ((index(visit,
      "_MONTH1_") and index(visit, 'POSTVAX3')=0 and index(visit, 'POSTVAX4')=0 and
      cohortn ne 1.16) or (visit="V7_MONTH1_S_R") and cohortn=1.16)))
      sv(keep=usubjid visit svstdt cohortn rename=(svstdt=m1pv3dt)
      where=(index(visit, "MONTH1") and index(visit, "POSTVAX3"))) sv(keep=usubjid
      visit svstdt cohortn rename=(svstdt=m1pv4dt) where=(index(visit, "MONTH1")
      and index(visit, "POSTVAX4"))) sv(keep=usubjid visit svstdt cohortn
      rename=(svstdt=w3pv1dt) where=((cohortn ne 1.16 and visit='V4_WEEK3_VAX2_S')
```

```
    or (cohortn eq 1.16 and visit='V4_WEEK3_VAX2_S_R'))) dvout (keep=usubjid
    dvstdt safety efficacy immuno multiple siteexcld pop1pddt pop2pddt pop3pddt
    period1_pop3) pop5_2 incl3p(in=incl3p keep=usubjid) incl8p(in=incl8p
    keep=usubjid);
by usubjid;

if a;
attrib BLDV1FL label="Blood Sample Drawn before Vax 1" BLDV2FL
    label="Blood Sample Drawn 1 Week after Vax 1" BLDV3FL
    label="Blood Sample Drawn before Vax 2" BLDV4FL
    label="Blood Sample Drawn 1 Week after Vax 2" BLDV5FL
    label="Blood Sample Drawn 2 Weeks after Vax 2" BLDV6FL
    label="Blood Sample Drawn 1 Month after Vax 2" BLDV7FL
    label="Blood Sample Drawn 6 Months after Vax 2" BLDV1DT
    label="Blood Sample Date before Vax 1" format=date9. BLDV2DT
    label="Blood Sample Date 1 Week after Vax 1" format=date9. BLDV3ADT
    label="Additional Bld Sample Date 3W after Vax1" format=date9. BLDV4ADT
    label="Additional Bld Sample Date 4W after Vax1" format=date9. BLDV5ADT
    label="Additional Bld Sample Date 5W after Vax1" format=date9. BLDV6ADT
    label="Additional Bld Sample Date 7W after Vax1" format=date9. BLDV3DT
    label="Blood Sample Date before Vax 2" format=date9. BLDV4DT
    label="Blood Sample Date 1 Week after Vax 2" format=date9. BLDV5DT
    label="Blood Sample Date 2 Weeks after Vax 2" format=date9. BLDV6DT
    label="Blood Sample Date 1 Month after Vax 2" format=date9. BLDV7DT
    label="Blood Sample Date 6 Months after Vax 2" format=date9. INCL1FL
    label="Are eligible for the study at rand" INCL2FL
    label="Have received Vax 1 as randomized" INCL3FL
    label="Have valid and DTM immuno result 1" INCL4FL
    label="Have valid and DTM immuno result 2" INCL5FL
    label="Have BD within the timeframe 1" INCL6FL
    label="No important PD determined by clinician" INCL7FL
    label="Received 2 doses as rand within window" INCL8FL
    label="Have valid and DTM immuno result 3" INCL9FL
    label="Have valid immuno within timeframe 2" INCL10FL
    label="Unblinded after 1M post Dose 2 visit" INCL17FL
    label="Have N-Binding result PD2" EXCL1FL label="Exclusion Flag 1" EXCRIT1
    label="Exclusion Criterion 1" format=$200. EXCL2FL label="Exclusion Flag 2"
    EXCRIT2 label="Exclusion Criterion 2" format=$200. EXCL3FL
    label="Exclusion Flag 3" EXCRIT3 label="Exclusion Criterion 3" format=$200.
    EXCL4FL label="Exclusion Flag 4" EXCRIT4 label="Exclusion Criterion 4"
    format=$200. EXCL5FL label="Exclusion Flag 5" EXCRIT5
    label="Exclusion Criterion 5" format=$200. EXCL6FL label="Exclusion Flag 6"
    EXCRIT6 label="Exclusion Criterion 6" format=$200. RSEXSAF
    label="Reason for Exclusion from Safety Pop" format=$200. EXCL7FL
    label="Exclusion Flag 7" EXCRIT7 label="Exclusion Criterion 7" format=$200.
    EXCL8FL label="Exclusion Flag 8" EXCRIT8 label="Exclusion Criterion 8"
    format=$200. EXCL9FL label="Exclusion Flag 9" EXCRIT9
    label="Exclusion Criterion 9" format=$200. EXCL10FL label="Exclusion Flag 10"
    EXCRIT10 label="Exclusion Criterion 10" format=$200. EXCL17FL
    label="Exclusion Flag 17" EXCRIT17 label="Exclusion Criterion 17"
    format=$200. EVAL01FL label="Dose 1 evaluable Immun Popu Flag" EVAL02FL
    label="Dose 2 evaluable Immun Popu Flag" AAI01FL
    label="Dose 1 all-available Immun Popu Flag" AAI02FL
    label="Dose 2 all-available Immun Popu Flag" EVALEFFL
```

```
   label="Evaluable Efficacy Popu Flag" EV14EFFL
   label="Evaluable Efficacy Popu Flag - 14 days" EV1MEFFL
   label="Evaluable Efficacy Popu Flag - 1 month" VASEFFL
   label="Evaluable Efficacy Pop Flag - asympt" EVSCEFFL
   label="Evaluable Efficacy Pop Flag - seroconv" AAI1EFFL
   label="Dose 1 all-available Efficacy Popu Flag" AAI2EFFL
   label="Dose 2 all-available Efficacy Popu Flag" AAI2ASFL
   label="Dose 2 all-available Asympt Popu Flag";

if not missing(pop5_txt) then
   RSEXSAF=strip(pop5_txt);
else if not missing(safety) then
   RSEXSAF="Did not provide informed consent";

if not missing(covis60765) then
   BLDV1FL="Y";
else
   BLDV1FL="N";

if not missing(covis60750) then
   BLDV2FL="Y";
else
   BLDV2FL="N";

if not missing(covis60751) and cohortn^=1.16 then
   BLDV3FL="Y";
else if not missing(covis1165454) and cohortn=1.16 then
   BLDV3FL="Y";
else if not missing(covis75849038) then
   BLDV3FL="Y";
else
   BLDV3FL="N";

if not missing(covis60752) and cohortn^=1.16 then
   BLDV4FL="Y";
else if not missing(covis1165455) and cohortn=1.16 then
   BLDV4FL="Y";
else if not missing(covis75849039) then
   BLDV4FL="Y";
else
   BLDV4FL="N";

if not missing(covis60753) and cohortn^=1.16 then
   BLDV5FL="Y";
else if not missing(covis1165456) and cohortn=1.16 then
   BLDV5FL="Y";
else
   BLDV5FL="N";

if not missing(covis60767) then
   BLDV6FL="Y";
else
   BLDV6FL="N";
```

```
if not missing(covis60768) then
   BLDV7FL="Y";
else
   BLDV7FL="N";

if not missing(codt60765) then
   BLDV1DT=codt60765;

if not missing(codt60750) then
   BLDV2DT=codt60750;

if not missing(codt60751) and cohortn=1.16 then
   BLDV3ADT=codt60751;
else if not missing(codt60751) then
   BLDV3DT=codt60751;
else if not missing(codt75849038) then
   BLDV3DT=codt75849038;

if not missing(codt1165454) and cohortn=1.16 then
   BLDV3DT=codt1165454;

if not missing(codt60752) and cohortn=1.16 then
   BLDV4ADT=codt60752;
else if not missing(codt60752) then
   BLDV4DT=codt60752;
else if not missing(codt75849039) then
   BLDV4DT=codt75849039;

if not missing(codt1165455) and cohortn=1.16 then
   BLDV4DT=codt1165455;

if not missing(codt60753) and cohortn=1.16 then
   BLDV5ADT=codt60753;
else if not missing(codt60753) then
   BLDV5DT=codt60753;

if not missing(codt1165456) and cohortn=1.16 then
   BLDV5DT=codt1165456;

if not missing(codt60754) and cohortn=1.16 then
   BLDV6ADT=codt60754;
else if not missing(codt60754) then
   BLDV6DT=codt60754;
else if not missing(codt75849040) then
   BLDV6DT=codt75849040;

if not missing(codt60767) then
   BLDV6DT=codt60767;

if not missing(codt60768) then
   BLDV7DT=codt60768;

if phasen ne 1 then
   do;
```

```
      if not missing(BLDV6DT) then
         v3date=BLDV6DT;
      else if not missing(v3p2) then
         v3date=input(v3p2, e8601da.);
      else
         v3date=vax102dt + 42;
   end;
else
   do;

      if not missing(BLDV4DT) then
         v3date=BLDV4DT;
      else if not missing(v5p1) then
         v3date=input(v5p1, e8601da.);
      else
         v3date=vax102dt + 8;
   end;
format v3date is8601da.;

if not missing(safety) or index(RSEXSAF, 'Safety') then
   SAFFL="N";

if RFICDT>. and RANDFL="Y" and ARM ne "SCREEN FAILURE" and INEX="" then
   INCL1FL="Y";
else
   INCL1FL="N";

if (vax101dt>. or vax102dt>.) and randfl="Y" and ARM ne ""
   and (((index(upcase(vax101), "BNT162B1") and index(upcase(arm), "BNT162B1"))
   or (index(upcase(vax101), "BNT162B2") and index(upcase(arm), "BNT162B2"))
   or (index(upcase(vax101), "PLACEBO") and index(upcase(arm), "PLACEBO")))
   or (vax101dt=. and vax102dt>. and ((index(upcase(vax102), "BNT162B1") and
   index(upcase(arm), "BNT162B1")) or (index(upcase(vax102), "BNT162B2") and
   index(upcase(arm), "BNT162B2")) or (index(upcase(vax102), "PLACEBO") and
   index(upcase(arm), "PLACEBO"))))) then
      INCL2FL="Y";
else
   INCL2FL="N";

if incl3p and VAX101DT>. and phasen=1 then
   INCL3FL="Y";
else if phasen=1 then
   INCL3FL="N";

if not missing(BLDV6DT) then
   visit3dt=BLDV6DT;

if not missing(safety) then
   INCL6FL="N";
else if (not missing(efficacy) and not(. < visit3dt < pop2pddt)
   and ((missing(vax102dt) or (. <pop2pddt <=vax102dt)) or (not
   missing(vax102dt) and pop2pddt-vax102dt<14) or (not missing(vax102dt) and
   pop2pddt-vax102dt>=14 and (pop2pddt<=visit3dt)) )) then
```

```
      INCL6FL="N";
else if not missing(immuno) then
   do;

      if index(trt02p, 'Dose')=0 then
         do;

            if missing(period1_pop3) then
               INCL6FL="Y";
            else
               do;

                  if . < v3date < POP3PDDT then
                     INCL6FL="Y";
                  else
                     INCL6FL="N";
               end;
         end;
   end;
else
   INCL6FL="Y";

if 19 <=VAX102DT-VAX101DT <=42 and vax101=vax102 and ARM ne ""
   and ((index(upcase(vax102), "BNT162B1") and index(upcase(arm), "BNT162B1"))
   or (index(upcase(vax102), "BNT162B2") and index(upcase(arm), "BNT162B2"))
   or (index(upcase(vax102), "PLACEBO") and index(upcase(arm), "PLACEBO"))) then
      INCL7FL="Y";
else
   INCL7FL="N";

if incl8p and VAX102DT>. then
   INCL8FL="Y";
else
   INCL8FL="N";
*1mpd2 after dose 3 will be exclude;

if UNBLNDDT>. and vax102dt>. and UNBLNDDT>visit3dt>. then
   INCL10FL="Y";
else if vax102dt>. and (.<UNBLNDDT<=visit3dt or .<UNBLNDDT<vax102dt+14) then
   INCL10FL="N";

if INCL3FL="N" and (VAX101DT>. or vax102dt>.) then
   do;
      EXCL3FL="Y";
      EXCRIT3="did not have at least 1 valid and determinate immunogenicity result after Dose 1 but before Dose 2";
   end;

if INCL8FL="N" and vax102dt>. then
   do;
      EXCL8FL="Y";
      EXCRIT8="did not have at least 1 valid and determinate immunogenicity result after Dose 2";
   end;

if INCL10FL="N" then
```

```
      do;
        EXCL10FL="Y";

        if .<UNBLNDDT<vax102dt+7 then
          EXCRIT10="unblinded prior to 7 days post Dose 2";
        else if vax102dt+7<=UNBLNDDT<vax102dt+14 then
          EXCRIT10="unblinded on or after 7 days but prior to 14 days post Dose 2";
        else if vax102dt+14<=UNBLNDDT<=visit3dt then
          EXCRIT10="unblinded on or after 14 days but no later than 1 month post Dose 2 visit";
      end;

    if INCL1FL="N" then
      do;
        EXCL1FL="Y";
        EXCRIT1="not eligible for the study at randomization";
      end;
    else
      do;

        if INCL2FL="N" then
          do;
            EXCL2FL="Y";
            EXCRIT2="did not receive Dose 1 as randomized";
          end;
        else
          do;

            if INCL7FL="N" then
              do;
                EXCL7FL="Y";
                EXCRIT7="did not receive all vaccination(s) as randomized or did not receive Dose 2 within the
predefined window (19-42 days after Dose 1)";
              end;
          end;
      end;

    if INCL6FL="N" then
      do;
        length _ttt_ $200;

        if not missing(safety) then
          _ttt_="Safety";

        if not missing(efficacy) then
          do;

            if not missing(vax102dt) then
              do;

                if .<pop2pddt-vax102dt<7 then
                  do;

                    if not missing(_ttt_) then
                      _ttt_=strip(_ttt_)||", Efficacy (within 7 days post Dose 2)";
```

```
            else
               _ttt_="Efficacy (within 7 days post Dose 2)";
            end;
         else if 7<=pop2pddt-vax102dt<14 then
            do;

               if not missing(_ttt_) then
                  _ttt_=strip(_ttt_)||", Efficacy (between 7-14 days post Dose 2)";
               else
                  _ttt_="Efficacy (between 7-14 days post Dose 2)";
            end;
         else if pop2pddt-vax102dt>=14 and (.<pop2pddt<=visit3dt) then
            do;

               if not missing(_ttt_) then
                  _ttt_=strip(_ttt_)||", Efficacy (between 14 days - 1 month post Dose 2)";
               else
                  _ttt_="Efficacy (between 14 days - 1 month post Dose 2)";
            end;
      end;
   end;

if not missing(immuno) then
   do;

      if . < POP3PDDT <=BLDV3DT then
         do;

            if not missing(_ttt_) then
               _ttt_=strip(_ttt_)||", Immunogenicity (on or before 3 weeks post Dose 1)";
            else
               _ttt_="Immunogenicity (on or before 3 weeks post Dose 1)";
         end;
      else if missing(BLDV3DT) and . < POP3PDDT <=w3pv1dt then
         do;

            if not missing(_ttt_) then
               _ttt_=strip(_ttt_)||", Immunogenicity (on or before 3 weeks post Dose 1)";
            else
               _ttt_="Immunogenicity (on or before 3 weeks post Dose 1)";
         end;
      else if missing(BLDV3DT) and missing(w3pv1dt) and not missing(vax101dt)
         and . < POP3PDDT <=(vax101dt + 23) then
            do;

            if not missing(_ttt_) then
               _ttt_=strip(_ttt_)||", Immunogenicity (on or before 3 weeks post Dose 1)";
            else
               _ttt_="Immunogenicity (on or before 3 weeks post Dose 1)";
         end;
      else if missing(vax101dt) then
         do;

            if not missing(_ttt_) then
```

```
                _ttt_=strip(_ttt_)||", Immunogenicity (on or before 3 weeks post Dose 1)";
            else
                _ttt_="Immunogenicity (on or before 3 weeks post Dose 1)";
          end;
        else if . < POP3PDDT <=v3date then
          do;

            if phasen ne 1 then
              do;

                if not missing(_ttt_) then
                  _ttt_=strip(_ttt_)||", Immunogenicity (on or before 1 month post Dose 2)";
                else
                  _ttt_="Immunogenicity (on or before 1 month post Dose 2)";
              end;
            else
              do;

                if not missing(_ttt_) then
                  _ttt_=strip(_ttt_)||", Immunogenicity (on or before 1 week post Dose 2)";
                else
                  _ttt_="Immunogenicity (on or before 1 week post Dose 2)";
              end;
          end;
        else if missing(v3date) and not missing(period1_pop3) then
          do;

            if phasen ne 1 then
              do;

                if not missing(_ttt_) then
                  _ttt_=strip(_ttt_)||", Immunogenicity (on or before 1 month post Dose 2)";
                else
                  _ttt_="Immunogenicity (on or before 1 month post Dose 2)";
              end;
            else
              do;

                if not missing(_ttt_) then
                  _ttt_=strip(_ttt_)||", Immunogenicity (on or before 1 week post Dose 2)";
                else
                  _ttt_="Immunogenicity (on or before 1 week post Dose 2)";
              end;
          end;
      end;
    EXCL6FL="Y";

    if not missing(_ttt_) then
      EXCRIT6="had important protocol deviation(s) as determined by the clinician for "||strip(_ttt_)||"
Population(s)";
    else
      EXCRIT6="";
  end;
```

```
if phasen=1 and randfl="Y" and saffl="Y" and (VAX101DT>. or vax102dt>.) and
   INCL3fl="Y" and index(RSEXSAF, 'Immunogenicity')=0 then
      AAI01FL="Y";
else if phasen=1 then
   AAI01FL="N";

if randfl="Y" and saffl="Y" and vax102dt>. and INCL8fl="Y" and index(RSEXSAF,
   'Immunogenicity')=0 then
      AAI02FL="Y";
else
   AAI02FL="N";

if RFICDT>. and RANDFL="Y" and ARM ne "SCREEN FAILURE" and INCL2fl="Y"
   and (INCL7fl="Y" and (vax10udt=. or (vax10udt>vax102dt>. and
   vax10udt>=vax102dt+7))) and
   VAX102DT>. and (UNBLNDDT=. or (UNBLNDDT>=vax102dt+7>.)) and saffl="Y" and
   INCL1FL="Y" and not (not missing(efficacy) and pop2pddt-vax102dt<7) and
   index(RSEXSAF, 'Efficacy')=0 then
      EVALEFFL="Y";
else
   EVALEFFL="N";

if RFICDT>. and RANDFL="Y" and ARM ne "SCREEN FAILURE" and INCL2fl="Y"
   and (INCL7fl="Y" and (vax10udt=. or (vax10udt>vax102dt>. and
   vax10udt>=vax102dt+14))) and
   VAX102DT>. and (UNBLNDDT=. or (UNBLNDDT>=vax102dt+14>.)) and saffl="Y" and
   INCL1FL="Y" and not (not missing(efficacy) and pop2pddt-vax102dt<14) and
   index(RSEXSAF, 'Efficacy')=0 then
      EV14EFFL="Y";
else
   EV14EFFL="N";

if RFICDT>. and RANDFL="Y" and ARM ne "SCREEN FAILURE" and INCL2fl="Y"
   and (INCL7fl="Y" and (vax10udt=. or (vax10udt>visit3dt>.))) and
   VAX102DT>. and (UNBLNDDT=. or (UNBLNDDT>visit3dt>.)) and saffl="Y" and
   INCL1FL="Y" and V3C19NIG ne "" and INCL10FL^="N" and not (not
   missing(efficacy) and pop2pddt<=visit3dt) and index(RSEXSAF, 'Efficacy')=0
   then
      EV1MEFFL="Y";
else
   EV1MEFFL="N";

if randfl="Y" and saffl="Y" and (vax101dt>. or vax102dt>.) and index(RSEXSAF,
   'Efficacy')=0 then
      AAI1EFFL="Y";
else
   AAI1EFFL="N";

if randfl="Y" and saffl="Y" and vax101dt>. and
   vax102dt>. and (UNBLNDDT=. or (UNBLNDDT>=vax102dt+7>.)) and index(RSEXSAF,
   'Efficacy')=0 then
      AAI2EFFL="Y";
else
   AAI2EFFL="N";
```

```
   if randfl="Y" and saffl="Y" and vax101dt>. and vax102dt>. and index(RSEXSAF,
      'Efficacy')=0 then
         AAI2ASFL="Y";
   else
      AAI2ASFL="N";
run;

*update criteria flag 9;

data incl9p_p2;
   set isva;

   if istestcd^="C19NIG" and isorres not in ("" "IND" "QNS" "NOT DONE");

   if (28 <=isdate - vax102dt <=42 and phasen in (2 3 4))
      or (6 <=isdate - vax102dt <=8 and phasen=1);

proc sort nodupkey;
   by usubjid;
run;

proc sort data=adsl;
   by usubjid;
run;

data adsl(drop=INCL9FL_p2 EXCL9FL_p2 EXCRIT9_p2 EVAL02FL_p2);
   length EXCRIT9_p2 $200.;
   merge adsl(in=a) incl9p_p2(keep=usubjid in=incl9p2);
   by usubjid;

   if a;

   if incl9p2 and vax102dt >. then
      INCL9FL_p2='Y';
   else
      INCL9FL_p2='N';

   if INCL9FL_p2="N" and vax102dt>. and incl1fl='Y' and incl2fl='Y' then
      do;
         **hup02 updated to be consistent with BLA;
         EXCL9FL_p2="Y";

         if phasen in (2 3 4) then
            EXCRIT9_p2="did not have at least 1 valid and determinate immunogenicity result within 28-42 days after
Dose 2";
         else if phasen=1 then
            EXCRIT9_p2="did not have at least 1 valid and determinate immunogenicity result within 6-8 days after
Dose 2";
      end;

   if INCL1fl="Y" and INCL2fl="Y" and (INCL7fl="Y" and vax10udt=.) and saffl="Y"
      and INCL9FL_p2="Y" and not((. < POP3PDDT <=v3date) or (v3date=. and not
      missing(period1_pop3)) ) and index(RSEXSAF, 'Immunogenicity')=0 then
```

```
        EVAL02FL_p2="Y";
    else
       EVAL02FL_p2="N";
    INCL9FL=INCL9FL_p2;
    EXCL9FL=EXCL9FL_p2;
    EXCRIT9=EXCRIT9_p2;
    EVAL02FL=EVAL02FL_p2;
    label EVAL02FL="Dose 2 evaluable Immun Popu Flag";
run;

*Update criteria flag 4;

data incl4p_re;
    set isva;

    if istestcd^="C19NIG" and isorres not in ("" "IND" "QNS" "NOT DONE");

    if 19 <=isdate - vax101dt <=23;
run;

proc sort data=incl4p_re nodupkey;
    by usubjid;
run;

proc sort data=adsl;
    by usubjid;
run;

data adsl;
    merge adsl(in=a) incl4p_re(keep=usubjid in=incl4);
    by usubjid;

    if a;
    INCL4FL='';
    EXCL4FL='';
    EXCRIT4='';
    EVAL01FL='';

    if phasen=1 then
       do;

          if incl4 and vax101dt >. then
             INCL4FL='Y';
          else
             INCL4FL='N';

          if INCL4FL='N' and vax101dt>. and INCL1FL='Y' and INCL2FL='Y' then
             do;
                EXCL4FL="Y";
                EXCRIT4="did not have at least 1 valid and determinate immunogenicity result within 19-23 days after
Dose 1";
             end;

          if phasen=1 and INCL1fl="Y" and INCL2fl="Y" and INCL4fl="Y" and saffl='Y'
```

```
            and incl6fl='Y' and index(RSEXSAF, 'Immunogenicity')=0 and
            not((.<pop3pddt <=bldv3dt) or (missing(bldv3dt) and .<pop3pddt <=w3pv1dt)
            or (missing(bldv3dt) and missing(w3pv1dt) and .<pop3pddt <=(vax101dt + 23))
            ) then
               EVAL01FL="Y";
         else if phasen=1 then
            EVAL01FL="N";
      end;
   attrib INCL4FL label="Have valid immuno within timeframe 1" EVAL01FL
      label="Dose 1 evaluable Immun Popu Flag";

proc sort;
   by usubjid;
run;

*Update criteria flag 17;
*not consider any post N-binding results which are after V301 ;

proc sql;
   create table isva_tmp as select a.*, b.v301dt from isva a left join (select
      usubjid, input(svstdtc, yymmdd10.) as v301dt from dataprot.sv where
      visit='V301_VAX3') b on a.usubjid=b.usubjid;
quit;

data incl17p(keep=usubjid);
   set isva_tmp;

   if istestcd="C19NIG" and isorres not in ("" "IND" "QNS" "NOT DONE");

   if not missing(vax102dt) and isdate > vax102dt;

   if . < v301dt < isdate then
      delete;

proc sort nodupkey;
   by usubjid;
run;

data nbind1st_0;
   set isva_tmp;

   if istestcd="C19NIG" and isorres not in ("" "IND" "QNS" "NOT DONE");

   if not missing(vax102dt) and isdate > vax102dt;

   if . < v301dt < isdate then
      delete;

proc sort;
   by usubjid isdate;
run;

data nbind1st(keep=usubjid isdate);
   set nbind1st_0;
```

```
   by usubjid isdate;

   if first.usubjid then
      output;
run;

data asy_usub(keep=usubjid SURVICDT);
   set dataprot.sv;

   if visit='V201_SURVEIL_CONSENT';

   if not missing(svstdtc);
   SURVICDT=input(svstdtc, e8601da.);
   format SURVICDT date9.;

proc sort nodupkey;
   by usubjid;
run;

data adsl(drop=nbind1stdt);
   attrib EVASEFFL label="Evaluable Efficacy Pop Flag - Asympt" EVSCEFFL
      label="Evaluable Efficacy Pop Flag - Seroconv" EXCL17FL
      label="Exclusion Flag 17" INCL18FL
      label="No PD prior to 1st N-Binding test PD2" EXCL18FL
      label="Exclusion Flag 18" EXCRIT18 label="Exclusion Criterion 18"
      format=$200. INCL9FL label="Have valid immuno within timeframe 2" INCL17FL
      label="Have N-Binding result PD2" EXCL4FL label="Exclusion Flag 4" EXCL9FL
      label="Exclusion Flag 9" SURVICDT
      label="Asymptomatic Surveillance Consented Date";
   merge adsl(in=a) incl17p(in=in17) asy_usub(in=asy)
      nbind1st(rename=(isdate=nbind1stdt));
   by usubjid;

   if a;

   if in17 then
      INCL17FL='Y';
   else
      INCL17FL='N';

   if incl17fl ne 'Y' and incl1fl='Y' then
      do;
         **hup02 updated to have crit17 exclusive with crit1;
         EXCL17FL='Y';
         EXCRIT17='did not have N-binding test result at a post(*ESC*){unicode 2013}Dose 2 visit';
      end;

   if not(. < pop2pddt < nbind1stdt) then
      incl18fl='Y';
   else
      incl18fl='N';

   if incl18fl ne 'Y' and nbind1stdt ne . then
      do;
```

```
        excl18fl='Y';
        excrit18='had important protocol deviation(s) as determined by the clinician for Efficacy (prior to the first
post(*ESC*){unicode 2013}Dose 2 N-binding antibody test) Population';
      end;

   if saffl='Y' and index(RSEXSAF, 'Efficacy')=0 and INCL1FL='Y' and INCL7fl="Y"
      and not(. < vax10udt <=SURVICDT) and not missing(SURVICDT) and
      not(. < pop2pddt <=SURVICDT) then
         EVASEFFL='Y';
   else
      EVASEFFL='N';

   if saffl='Y' and index(RSEXSAF, 'Efficacy')=0 and INCL1FL='Y' and INCL7fl="Y"
      and not (. < vax10udt <=nbind1stdt) and INCL17FL='Y' and incl18fl='Y' then
         EVSCEFFL='Y';
   else
      EVSCEFFL='N';
run;

*****************************************************************************************;
* Specification 8 *;
* OTHER POP SELECTION FLAGS *;
* 1 - Determine population flags. *;
* 2 - Read in flags - PROCGR1/PROCGR1N. *;
* 3 - Read in flags - PEDIMMFL. *;
* 4 - PC1MD2FL. *;
* 5 - HIV flag. *;
* 6 - Determine subjects with booster dose. *;
*****************************************************************************************;

data adsl;
   merge adsl(in=a) sv(in=b keep=usubjid visit where=(index(visit, "V104_")=0 and
      index(visit, "_MONTH6_")));
   label SCREEN="Screening" DS3KFL="FU to 6MPD2"
      DS30KFL="Phase 3 30k Subjects Flag"
      OPBOUFL="Subjects Received Placebo & unblinded" JPNFL="Japanese Subject Flag"
      PEDREAFL="Phase 2/3 Pop for 12-25 Reacto Subset"
      STEXCFL="Site/Subject Exclusion Flag for SQE"
      UNKRDFL="Unknown Randomization Group Flag";
   by usubjid;

   if a;

   if (tr02sdt>. or UNBLNDDT>.) and actarm in ("Placebo") then
      OPBOUFL="Y";
   else
      OPBOUFL="N";

   if phasen ne 1 and vax101="BNT162b2 (30 (*ESC*){unicode 03BC}g)" and
      vax102="BNT162b2 (30 (*ESC*){unicode 03BC}g)" and
      trt01p='BNT162b2 Phase 2/3 (30 mcg)' then
         DS3KFL="Y";
   else
      DS3KFL="N";
```

```
    if not missing(RFICDT) then
       SCREEN='Y';
    else
       SCREEN="N";

    if '27JUL2020'd<=rficdt and .<randdt<='09OCT2020'd and phasen ne 1 then
       DS30KFL="Y";
    else
       DS30KFL="N";

    if RACIALD="JAPANESE" then
       JPNFL="Y";
    else
       JPNFL="N";

    if reactofl="Y" and phasen ne 1 and agegr4n in (1 2) then
       PEDREAFL="Y";
    STEXCFL="";

    if arm="" and randdt>. then
       UNKRDFL="Y";
run;

*Add PEDIMMFL for pediatric info;

proc import datafile="&expath./c4591001-subject-list-for-12-25-immuno-analysis-27jan2021.xlsx"
       out=pop12_25 dbms=xlsx replace;
    getnames=yes;
run;

proc sort data=pop12_25;
    by usubjid;
run;

data adsl;
    merge adsl(in=a) pop12_25(in=b);
    by usubjid subjid;

    if b then
       PEDIMMFL='Y';

    if a;
    label PEDIMMFL="Pop for Non-inferiority Assessement";
run;

*HIV subject;

proc import datafile="&expath./201114-hiv-preferred-terms.xlsx" out=hivpt
       dbms=xlsx replace;
    getnames=yes;
run;

proc sort;
```

```
   by term;
run;

proc sort data=dataprot.mh out=mh_hiv (keep=usubjid mhdecod
      rename=mhdecod=term);
   by mhdecod;
run;

data hiv1;
   merge mh_hiv (in=a) hivpt (in=b);
   by term;

   if a and b;
run;

proc sort;
   by usubjid;
run;

data adsl;
   merge adsl (in=a) hiv1 (in=b keep=usubjid);
   by usubjid;

   if a;
   ***** Flag for HIV +ve Subjects *****;

   if a and b then
      HIVFL="Y";
   else
      HIVFL="N";
   label HIVFL="HIV Positive Subjects Flag";
   ***** Set all Efficacy Flags to N for Phase 1 subjects *****;
run;

/**** START - Setting up ADSYMPT dataset *****/;
** Get FA data. **;

proc sort data=dataprot.face(keep=studyid domain usubjid faseq fatestcd fatest
      faobj facat fascat faorres fastresc fadrvfl visitnum visit fadtc) out=face;
   by usubjid visitnum visit fatestcd faobj faorres;
   where upcase(strip(facat))='EFFICACY';
run;

data face1 face_stdt(keep=usubjid faorres visitnum visit
      rename=(faorres=fastdtc)) face_endt(keep=usubjid faorres visitnum visit
      rename=(faorres=faendtc)) face_ong(keep=usubjid faorres visitnum visit
      rename=(faorres=faong));
   set face;
   by usubjid visitnum visit fatestcd faobj faorres;

   if upcase(strip(fatestcd))='FSYMDATE' then
      output face_stdt;
   else if upcase(strip(fatestcd))='LSYMDATE' then
      output face_endt;
```

```
    else if upcase(strip(fatestcd))='SYMONGO' then
       output face_ong;
    else
       output face1;
run;

data face2;
    merge face1(in=a) face_stdt(in=b) face_endt(in=c) face_ong(in=d);
    by usubjid visitnum visit;

    if a;
run;

data fa(keep=studyid domain usubjid paramn paramcd param parcat1 parcat2 aval
       avalc adt astdt aendt visitnum visit) fa_excluded;
    set face2;
    length paramn 8 paramcd $8 param parcat1 parcat2 avalc $200;
    param=upcase(strip(faobj));
    parcat1='SIGNS AND SYMPTOMS OF DISEASE';
    parcat2='RESPIRATORY ILLNESS';
    avalc=strip(fastresc);

    if strip(param) in ('CHILLS', 'DIARRHEA', 'FEVER') then
       do;
          paramcd=strip(param);

          if paramcd='CHILLS' then
             paramn=1;

          if paramcd='DIARRHEA' then
             paramn=2;

          if paramcd='FEVER' then
             paramn=3;
       end;
    else if strip(param)='NEW LOSS OF TASTE OR SMELL' then
       do;
          paramn=4;
          paramcd='NLTSTSML';
       end;
    else if strip(param)='NEW OR INCREASED COUGH' then
       do;
          paramn=5;
          paramcd='NCOUG';
       end;
    else if strip(param)='NEW OR INCREASED MUSCLE PAIN' then
       do;
          paramn=6;
          paramcd='NMUSPN';
       end;
    else if strip(param)='NEW OR INCREASED SHORTNESS OF BREATH' then
       do;
          paramn=7;
          paramcd='NSTBRTH';
```

```
   end;
else if strip(param)='NEW OR INCREASED SORE THROAT' then
   do;
      paramn=8;
      paramcd='NSRTHROT';
   end;
else if strip(param)='VOMITING' then
   do;
      paramn=9;
      paramcd='VOMIT';
   end;
else if strip(param)='LOSS OF TASTE/SMELL' then
   do;
      paramn=10;
      paramcd='LSTSTSML';
   end;
else if strip(param) in ('NEW OR INCREASED NASAL CONGESTION',
   'NASAL CONGESTION') then
      do;
      paramn=11;
      paramcd='NNSLCONG';
      param='NEW OR INCREASED NASAL CONGESTION';
   end;
else if strip(param)='NEW OR INCREASED NASAL DISCHARGE' then
   do;
      paramn=12;
      paramcd='NNSLDSCH';
   end;
else if strip(param)='NEW OR INCREASED SPUTUM PRODUCTION' then
   do;
      paramn=13;
      paramcd='SPUTPROD';
   end;
else if strip(param) in ('NEW OR INCREASED WHEEZING', 'WHEEZING') then
   do;
      paramn=14;
      paramcd='WHEEZ';
      param='NEW OR INCREASED WHEEZING';
   end;
else if strip(param)='FATIGUE' then
   do;
      paramn=15;
      paramcd='FATIGUE';
      param='FATIGUE';
   end;
else if strip(param)='HEADACHE' then
   do;
      paramn=16;
      paramcd='HEADACHE';
      param='HEADACHE';
   end;
else if strip(param)='NAUSEA' then
   do;
      paramn=18;
```

```
            paramcd='NAUSEA';
            param='NAUSEA';
        end;
    else
        do;
            id=prxparse('/' || 'RUNNY NOSE' || '/i');
            call prxsubstr(id, param, point, lng);

            if lng > 0 or upcase(faobj)='RHINORRHOEA' then
                do;
                    paramn=17;
                    paramcd='RIHNRA';
                    param='RHINORRHOEA';
                end;
        end;
    aval=.;
    adt=input(fadtc, ?? yymmdd10.);
    astdt=input(fastdtc, ?? yymmdd10.);
    aendt=input(faendtc, ?? yymmdd10.);
    format adt astdt aendt date9.;

    if not (strip(reverse(substr(reverse(strip(visit)), 1, 3))) in ('1_S', '2_S',
        'S_R', '4_S', '6_S', '_NS', '4_L', '6_L', 'SCR') or strip(visit)
        in ('V3_MONTH1_POSTVAX2_L', 'V5_MONTH12_L')) then
            do;

            if paramcd ^='' then
                output fa;
            else
                output fa_excluded;
        end;
run;

proc sql;
    create table fa_prnt as select distinct faobj from fa_excluded where
        faobj ^='';
quit;

** Get IS data. **;

data is(keep=studyid domain usubjid paramn paramcd param parcat1 parcat2 aval
        avalc adt astdt aendt visitnum visit isspec ismethod);
    set dataprot.is;
    where strip(istestcd) in ('C19NIG');
    length paramn 8 paramcd $8 param parcat1 parcat2 avalc $200;
    parcat1=strip(iscat);
    parcat2='';
    paramn=90;
    paramcd=strip(istestcd);
    param=upcase(strip(istest));
    aval=.;
    avalc=upcase(strip(isorres));
    adt=input(isdtc, ?? yymmdd10.);
    astdt=.;
```

```
    aendt=.;
    format adt astdt aendt date9.;
    *if strip(visit) in ('V1_DAY1_VAX1_L') then output;
run;

** Get MB data. **;

data mb(keep=studyid domain usubjid paramn paramcd param parcat1 parcat2 aval
        avalc adt astdt aendt visitnum visit_ mbloc mbspec mbmethod
        rename=(visit_=visit));
    set dataprot.mb;
    where (upcase(strip(mbtest))='SEVERE ACUTE RESP SYNDROME CORONAVIRUS 2' and
        upcase(strip(mbmethod))="IMMUNOCHROMATOGRAPHY") or (upcase(strip(mbtest))
        in ('CEPHEID RT-PCR ASSAY FOR SARS-COV-2',
        'CEPHEID RT-PCR ASSAY OF SARS-COV-2') and
        upcase(strip(mbmethod))='REVERSE TRANSCRIPTASE PCR');
    length paramn 8 paramcd $8 param parcat1 parcat2 avalc $200 visit_ $64;

    if upcase(strip(mbtest))='SEVERE ACUTE RESP SYNDROME CORONAVIRUS 2' and
        strip(spdevid) not in ('34', '44', '68') then
            do;
            mborres='UNKNOWN';
            mbstresc='UNK';
        end;
    parcat1=strip(mbcat);
    parcat2='';

    if strip(mbtestcd)='SARSCOV2' then
        paramn=40;

    if strip(mbtestcd)='RTCOV2NS' then
        paramn=41;
    paramcd=strip(mbtestcd);
    param=upcase(strip(mbtest));
    aval=.;
    avalc=strip(mborres);
    visit_=strip(visit);
    adt=input(mbdtc, ?? yymmdd10.);
    astdt=.;
    aendt=.;
    format adt astdt aendt date9.;

    if not (strip(reverse(substr(reverse(strip(visit)), 1, 3))) in ('1_S', '2_S',
        'S_R', '4_S', '6_S', '_NS', '4_L', '6_L', 'SCR') or strip(visit)
        in ('V3_MONTH1_POSTVAX2_L', 'V5_MONTH12_L')) then
            output;
run;

proc sort data=mb out=mb1 nodup;
    by usubjid paramn adt visitnum avalc;
run;

data adsympt1;
    set fa is mb1;
```

```
   avisitn=visitnum;
   avisit=strip(visit);
run;

proc sort data=adsympt1 out=adsympt2 nodup;
   by domain usubjid visitnum visit adt astdt aendt isspec ismethod mbloc
      mbmethod mbspec;
run;

%let __excl_vis1a =
%str('SCR','V1_DAY1_VAX1_S','V2_DAY2_POSTVAX1_S','V3_WEEK1_POSTVAX1_S','V4_WEEK3_VAX2_S','
V5_WEEK1_POSTVAX2_S','V6_WEEK2_POSTVAX2_S','V7_MONTH1_S');
%let __excl_vis1b =
%str('V4_WEEK3_VAX2_S_R','V5_WEEK1_POSTVAX2_S_R','V6_WEEK2_POSTVAX2_S_R','V7_MONTH1_S_
R','V8_MONTH6_S','V9_MONTH12_S','V10_MONTH24_S');
%let __excl_vis2 =
%str('V1_DAY1_VAX1_NS','V2_VAX2_NS','V3_WEEK2_POSTVAX2_NS','V4_MONTH1_NS','V5_MONTH6_NS'
,'V6_MONTH12_NS','V7_MONTH24_NS');
%let __excl_vis3 =
%str('V1_DAY1_VAX1_L','V2_VAX2_L','V3_MONTH1_POSTVAX2_L','V4_MONTH6_L','V5_MONTH12_L','V6
_MONTH24_L','POT_COVID_ILL','POT_COVID_CONVA');
** Get CE data. **;

data __ce(keep=usubjid domain adt astdt aendt visitnum visit);
   set dataprot.ce;
   where upcase(strip(cecat))='SEVERE COVID-19 ILLNESS' and upcase(strip(cescat))
      in ('SIGNIFICANT ACUTE RENAL DYSFUNCTION',
      'SIGNIFICANT ACUTE HEPATIC DYSFUNCTION',
      'SIGNIFICANT ACUTE NEUROLOGIC DYSFUNCTION');
   adt=input(cedtc, ?? yymmdd10.);
   astdt=input(cestdtc, ?? yymmdd10.);
   aendt=input(ceendtc, ?? yymmdd10.);
   format adt astdt aendt yymmdd10.;
run;

** Get FA data. **;

proc sort data=dataprot.face(keep=studyid usubjid domain faseq fatestcd fatest
      faobj facat fascat faorres fastresc fadrvfl visitnum visit fadtc) out=__face;
   by usubjid visitnum visit fatestcd faobj faorres;
   where upcase(strip(facat))='EFFICACY';
run;

data __face1 __face_stdt(keep=usubjid faorres visitnum visit
      rename=(faorres=fastdtc)) __face_endt(keep=usubjid faorres visitnum visit
      rename=(faorres=faendtc)) __face_ong(keep=usubjid faorres visitnum visit
      rename=(faorres=faong));
   set __face;
   by usubjid visitnum visit fatestcd faobj faorres;

   if upcase(strip(fatestcd))='FSYMDATE' then
      output __face_stdt;
   else if upcase(strip(fatestcd))='LSYMDATE' then
      output __face_endt;
```

```
   else if upcase(strip(fatestcd))='SYMONGO' then
      output __face_ong;
   else
      output __face1;
run;

data __fa(keep=usubjid domain adt astdt aendt visitnum visit);
   merge __face1(in=a) __face_stdt(in=b) __face_endt(in=c) __face_ong(in=d);
   by usubjid visitnum visit;

   if a;
   adt=input(fadtc, ?? yymmdd10.);
   astdt=input(fastdtc, ?? yymmdd10.);
   aendt=input(faendtc, ?? yymmdd10.);
   format adt astdt aendt yymmdd10.;
run;

** Get data from HO. **;

proc sql;
   create table __ho1 as select * from dataprot.ho left join (select qnam,
      qlabel, qval from dataprot.suppho as b where upcase(strip(qnam))='HCUHSP') on
      strip(usubjid)=strip(b.usubjid) and strip(put(hoseq,
      best.))=strip(b.idvarval);
   create table __ho2 as select * from __ho1 left join (select hostdtc as
      hostdtc_, hoendtc as hoendtc_, hoenrtpt as hoenrtpt_, hoentpt as hoentpt_
      from __ho1 as b where upcase(strip(hocat))='HOSPITALIZATION STATUS' and
      upcase(strip(hoterm))='HOSPITAL') on usubjid=b.usubjid and
      visitnum=b.visitnum and visit=b.visit and qnam ^='' order by usubjid, hoseq,
      hostdtc;
quit;

data __ho(keep=usubjid domain adt astdt aendt visitnum visit);
   set __ho2;
   adt=input(hodtc, ?? yymmdd10.);

   if upcase(strip(hoterm))='ICU' then
      do;
         astdt=input(hostdtc, ?? yymmdd10.);
         aendt=input(hoendtc, ?? yymmdd10.);
         output;
      end;

   if upcase(strip(qnam))='HCUHSP' then
      do;
         astdt=input(hostdtc_, ?? yymmdd10.);
         aendt=input(hoendtc_, ?? yymmdd10.);
         output;
      end;
   format adt astdt aendt yymmdd10.;
run;

** Get IS data. **;
```

```
data __is(keep=usubjid domain adt astdt aendt visitnum visit);
   set dataprot.is;
   where strip(istestcd) in ('C19NIG');
   adt=input(isdtc, ?? yymmdd10.);
   astdt=.;
   aendt=.;
   format adt astdt aendt yymmdd10.;
run;

** Get LB data. **;

data __lb(keep=usubjid domain adt astdt aendt visitnum visit_
      rename=(visit_=visit));
   set dataprot.lb;
   where upcase(strip(lbcat))='OXYGENATION PARAMETERS';
   length visit_ $64;
   visit_=strip(visit);
   adt=input(lbdtc, ?? yymmdd10.);
   astdt=.;
   aendt=.;
   format adt astdt aendt yymmdd10.;
run;

** Get MB data. **;

data __mb(keep=usubjid domain adt astdt aendt visitnum visit_
      rename=(visit_=visit));
   set dataprot.mb;
   where (upcase(strip(mbtest))='SEVERE ACUTE RESP SYNDROME CORONAVIRUS 2' and
      upcase(strip(mbmethod))="IMMUNOCHROMATOGRAPHY") or (upcase(strip(mbtest))
      in ('CEPHEID RT-PCR ASSAY FOR SARS-COV-2',
      'CEPHEID RT-PCR ASSAY OF SARS-COV-2') and
      upcase(strip(mbmethod))='REVERSE TRANSCRIPTASE PCR');
   length visit_ $64;
   visit_=strip(visit);
   adt=input(mbdtc, ?? yymmdd10.);
   astdt=.;
   aendt=.;
   format adt astdt aendt yymmdd10.;
run;

** Get VS data. **;

data __vs(keep=usubjid domain adt astdt aendt visitnum visit);
   set dataprot.vs;
   where upcase(strip(vscat))='GENERAL VITAL SIGNS' and strip(vstestcd)
      in ('RESP', 'HR', 'OXYSAT', 'DIABP', 'SYSBP');
   adt=input(vsdtc, ?? yymmdd10.);
   astdt=.;
   aendt=.;
   format adt astdt aendt yymmdd10.;
run;

data __visits_sdtm;
```

```
   set __ce __fa __ho __is __lb __mb __vs;
run;

proc sort data=__visits_sdtm nodup;
   by usubjid visitnum visit adt astdt aendt domain;
run;

data __visits_sdtm_rv1(drop=visit_) __visits_sdtm_rv1a(drop=visit_ visitnum
      covid_vis_cnt) __covid_vis_cnt(keep=usubjid covid_vis_cnt);
   set __visits_sdtm;
   by usubjid visitnum visit adt astdt aendt domain;
   visitnum_bak=visitnum;
   visit_bak=strip(visit);

   if length(visit) >=8 then
      do;

         if domain='MB' and substr(strip(visit), 8, 1) in ('1', '2', '3', '4', '5',
            '6', 'R') and substr(strip(visit), 1, 6)='COVID_' then
               rvflg=1;

         if rvflg=1 then
            visit=substr(visit, 1, 7);
      end;
   ** Create Covid visits count to be used for repeat visits. **;
   length visit_ $200;

   if first.usubjid then
      do;
         covid_vis_cnt=0;
         visit_='';
      end;

   if length(visit_bak) >=6 and upcase(substr(strip(visit_bak), 1, 6))='COVID_'
      and strip(visit_) ^=strip(visit_bak) and rvflg ^=1 then
         do;
         covid_vis_cnt=sum(covid_vis_cnt, 1);
         visit_=strip(visit_bak);
      end;

   if rvflg=1 then
      output __visits_sdtm_rv1a;
   else
      output __visits_sdtm_rv1;

   if last.usubjid then
      output __covid_vis_cnt;
   retain covid_vis_cnt visit_;
run;

proc sql;
   ** Get visitnums for repeat visits. **;
   create table __visits_sdtm_rv2a as select distinct * from (select * from
      __visits_sdtm_rv1a) left join (select visitnum from __visits_sdtm_rv1 as b
```

```
      where rvflg ^=1) on usubjid=b.usubjid and visit=b.visit;
   ** Check if any of them missing visitnum from above. **;
   create table __visits_sdtm_rv3a as select * from __visits_sdtm_rv2a left
      join (select visitnum as visitnum_rv, visit as visit_rv, astdt as astdt_rv,
      aendt as aendt_rv from __visits_sdtm_rv1 as b where domain='FA' and
      astdt ^=. and aendt ^=.) on usubjid=b.usubjid and b.astdt <=adt <=b.aendt and
      visitnum=.;
   ** Get visits count to assign visitnums. **;
   create table __visits_sdtm_rv4a as select * from __visits_sdtm_rv3a left
      join (select covid_vis_cnt from __covid_vis_cnt as b) on usubjid=b.usubjid
      order by domain, usubjid, visitnum, visit, adt, astdt, aendt;
quit;

data __visits_sdtm_rv5a;
   set __visits_sdtm_rv4a;
   by domain usubjid visitnum visit adt astdt aendt;

   if visitnum=. then
      do;

         if visitnum_rv ^=. and visit_rv ^='' then
            do;
               visitnum=visitnum_rv;
               visit=strip(visit_rv);
            end;
         else
            visitnum=sum(covid_vis_cnt, 1);
      end;
run;

data __visits_raw;
   set __visits_sdtm_rv1 __visits_sdtm_rv5a(drop=visitnum_rv visit_rv astdt_rv
      astdt_rv covid_vis_cnt);
run;

proc sort data=__visits_raw out=__visits_raw_unq nodupkey;
   by usubjid visitnum visit adt astdt aendt domain;
run;

data __visits_all;
   recseq=put(_n_, z7.);
   set __visits_raw_unq;

   if domain in ('IS', 'LB', 'MB', 'VS') then
      astdt=adt;
   *if domain = 'HO' and adt ^= . and astdt = . then astdt = adt;

   if strip(visit) not in (&__excl_vis1a, &__excl_vis1b, &__excl_vis2,
      &__excl_vis3) and visitnum ^=. and visit ^='' then
         visflg=1;
   else
      visflg=0;
run;
```

```
proc sort data=__visits_all out=__visits1(drop=) nodupkey;
   by usubjid astdt descending aendt visitnum visit;
   where visflg=1;
run;

proc sort data=__visits1 out=__visits_unq_vis1(keep=domain usubjid visitnum
      visit) nodupkey;
   by usubjid visitnum visit;
run;

** Check if an unplanned visit has FA records with date. **;

proc sort data=__visits_all out=__visits1_fa nodupkey;
   by usubjid visitnum visit;
   where domain='FA' and visflg=1;
run;

** When no FA visit is present, then exclude. **;

data __visits_unq_vis1_a(keep=usubjid visitnum visit eligflg);
   merge __visits_unq_vis1(in=a) __visits1_fa(in=b);
   by usubjid visitnum visit;

   if a and b then
      eligflg=1;
run;

data __visits_unq_vis2;
   set __visits_unq_vis1_a(where=(eligflg=1));
   by usubjid visitnum visit;

   if first.usubjid and last.usubjid then
      mlvisflg=0;
   else
      mlvisflg=1;
run;

proc sql;
   create table __visits2 as select * from __visits1 left join (select mlvisflg
      from __visits_unq_vis2 as b) on usubjid=b.usubjid and visitnum=b.visitnum
      order by usubjid, astdt, aendt desc, visitnum;
   ** For subjects that were not part of FA, combine their multiple different visits that have same start date into single
visit. **;
   ** Add such records to __visit2 data. **;
   create table __visits2a as select * from __visits2 left join (select distinct
      usubjid as usubjid_same_dt from (select * from (select * from __visits2 where
      mlvisflg ^=1) inner join (select astdt as astdt_same, visitnum as
      visitnum_not, visit as visit_not from __visits2 as b) on usubjid=b.usubjid
      and astdt=b.astdt and visitnum ^=b.visitnum and visit ^=b.visit) as b) on
      usubjid=b.usubjid order by usubjid, astdt, aendt desc, visitnum;
quit;

data __visits3(drop=mlvisflg usubjid_same_dt) __visits3a(drop=mlvisflg
      usubjid_same_dt clsp_pros_flg);
```

```
    set __visits2a;
    by usubjid astdt descending aendt visitnum;
    where mlvisflg=1 or usubjid_same_dt ^=";

    if (domain='FA') or (domain='HO' and astdt ^=. and aendt ^=.) or (domain='VS'
       and astdt ^=.) then
          do;
          clsp_pros_flg=1;
          output __visits3a;
       end;
    output __visits3;
run;

data __visits4 __visits4_clsp(keep=recseq usubjid visitnum visit astdt clspfl
       avisitn avisit);
    set __visits3a;
    nxtobs=_n_ + 1;
    by usubjid astdt descending aendt visitnum;

    if not last.usubjid then
       set __visits3a(keep=usubjid visitnum visit astdt aendt
          rename=(usubjid=usubjid_nxt visitnum=visitnum_nxt visit=visit_nxt
          astdt=astdt_nxt aendt=aendt_nxt)) point=nxtobs;

    if first.usubjid then
       do;
          astdt_=astdt;
          aendt_=aendt;
          visitnum_=visitnum;
          visit_=visit;
       end;

    if usubjid=usubjid_nxt then
       do;

          if resetflg='Y' then
             do;
                astdt_=astdt;
                aendt_=aendt;
                visitnum_=visitnum;
                visit_=visit;
                resetflg=";
             end;
          ** Check if nxt start is in range of current and expand the date range. **;

          if aendt_ ^=. and astdt_ <=astdt_nxt <=sum(aendt_, 3) then
             do;

                if aendt_ < astdt_nxt then
                   aendt_=astdt_nxt;

                if aendt_nxt ^=. and aendt_ < aendt_nxt then
                   aendt_=aendt_nxt;
             end;
```

```
        ** Check the current dates and visits and collapse. **;

        if visitnum_ ^=visitnum then
           do;

              if (aendt_=. and astdt_ <=astdt <=sum(astdt_, 3))
                 or (aendt_ ^=. and . < astdt_ <=astdt <=aendt_) then
                    do;
                    clspfl='Y';
                    avisitn=visitnum_;
                    avisit=visit_;
                 end;
           end;

        if aendt=. and astdt <=astdt_nxt <=sum(astdt, 3) then
           astdt_=astdt;
     end;
   ** Reset the _ vars with current visit.;

   if (aendt_=. and sum(astdt_, 3) < astdt_nxt) or (aendt_ ^=. and
      astdt_ < sum(aendt_, 3) < astdt_nxt) then
         resetflg='Y';
   output __visits4;

   if clspfl='Y' then
      output __visits4_clsp;
   format astdt aendt astdt_nxt aendt_nxt astdt_ aendt_ yymmdd10.;
   retain visitnum_ visit_ astdt_ aendt_ resetflg;
run;

proc sort data=__visits4_clsp out=__visits4_clsp_b nodupkey;
   by recseq usubjid visitnum visit astdt clspfl avisitn avisit;
run;

proc sql;
   create table __visits5 as select * from __visits3 left join (select astdt as
      astdt_c, clspfl, avisitn as avisitn_c, avisit as avisit_c from
      __visits4_clsp_b as b where clspfl='Y') on usubjid=b.usubjid
      and ((visitnum=b.visitnum and clsp_pros_flg=. and b.astdt <=astdt)
      or (recseq=b.recseq and clsp_pros_flg=1)) order by usubjid, astdt, aendt
      desc, visitnum, recseq, astdt_c;
quit;

data __visits6;
   set __visits5;
   by usubjid astdt descending aendt visitnum recseq astdt_c;

   if clspfl='Y' and avisitn=. then
      do;
         avisitn=avisitn_c;
         avisit=avisit_c;
      end;

   if avisitn=. then
```

```
      do;
         avisitn=visitnum;
         avisit=visit;
      end;

   if last.recseq then
      keepflg=1;
run;

** Prepare all visits. **;

data __visits_raw_prepare;
   set __visits_raw;

   if domain in ('IS', 'LB', 'MB', 'VS') then
      astdt=adt;

   if domain in ('IS', 'LB', 'MB', 'VS') then
      do;
         astdt=adt;
         adtflg=1;
      end;

   if strip(visit) not in (&__excl_vis1a, &__excl_vis1b, &__excl_vis2,
      &__excl_vis3) and astdt ^=. and visitnum ^=. and visit ^='' then
         visflg=1;
   else
      visflg=0;
run;

proc sql;
   create table __visits_all_1 as select * from __visits_all left join (select
      mlvisflg from __visits_unq_vis2 as b) on usubjid=b.usubjid and
      visitnum=b.visitnum;
   create table __visits_all_2 as select * from __visits_all_1 left join (select
      avisitn, avisit, clspfl from __visits6 as b where keepflg=1) on
      usubjid=b.usubjid and visitnum=b.visitnum and visit=b.visit and astdt=b.astdt
      and aendt=b.aendt;
   create table __visits_all_3 as select * from __visits_raw_prepare left
      join (select visflg as visflg_, mlvisflg, astdt as astdt_, aendt as aendt_,
      avisitn, avisit, clspfl from __visits_all_2 as b) on domain=b.domain and
      usubjid=b.usubjid and visitnum=b.visitnum and visit=b.visit and
      visitnum_bak=b.visitnum_bak and visit_bak=b.visit_bak and adt=b.adt and
      astdt=b.astdt and aendt=b.aendt order by usubjid, astdt_, aendt_ desc,
      visitnum;
quit;

data clsp_covid_vis_test clsp_covid_vis(drop=adtflg rvflg visitnum_bak
      visit_bak visflg_ visflg mlvisflg astdt_ aendt_);
   set __visits_all_3;
   by usubjid astdt_ descending aendt_ visitnum;

   if not(visflg=1 and mlvisflg=1) then
      do;
```

```
        avisitn=visitnum;
        avisit=strip(visit);
      end;

   if rvflg=1 then
      do;
        visitnum=visitnum_bak;
        visit=visit_bak;

        if avisitn=. then
          avisitn=1;
      end;
   output clsp_covid_vis_test;

   if adtflg=1 then
      astdt=.;

   if rvflg=1 then
      clspfl='Y';

   if visflg=1 or rvflg=1 then
      output clsp_covid_vis;
run;

** Report. **;

proc sql;
   create table __report1 as select distinct * from
          (select distinct * from clsp_covid_vis_test
      where strip(visit) not in (&__excl_vis1a, &__excl_vis1b, &__excl_vis2,
      &__excl_vis3)) inner join
          (select clspfl as clspfl_ from
      clsp_covid_vis_test as b where clspfl='Y') on usubjid=b.usubjid order by
      usubjid, astdt_, aendt_ desc, visitnum;
quit;

data __report2(drop=rvflg visitnum_bak visit_bak visflg_ visflg mlvisflg astdt_
      aendt_ clspfl_);
   set __report1;
   by usubjid astdt descending aendt visitnum;

   if adtflg=1 then
      astdt=.;
run;

**** Drop all records for Phase 1 subjects from ADSYMPT ****;

proc sql;
   create table adsympt3_all as select * from adsympt2 left join (select avisitn
      as avisitn_clsp, avisit as avisit_clsp, clspfl from clsp_covid_vis as b where
      clspfl='Y') on domain=b.domain and usubjid=b.usubjid and visitnum=b.visitnum
      and visit=b.visit and adt=b.adt and astdt=b.astdt and aendt=b.aendt inner
      join (select phasen, phase from adsl c where phasen >=2) on usubjid=c.usubjid
      order by usubjid, visitnum, visit, adt, astdt, aendt;
```

```
quit;

data adsympt4 adsympt (keep=usubjid visit: param: parcat: aval: adt astdt aendt
      avisit avisitn);
   recseq=put(_n_, z7.);
   set adsympt3_all;

   if clspfl='Y' then
      do;
         avisitn=avisitn_clsp;
         avisit=avisit_clsp;
      end;
   avalc=strip(avalc);

   if avalc='.' then
      avalc='';

   if avalc='UNKNOWN' then
      avalc='UNK';

   if avalc='POSITIVE' then
      avalc='POS';

   if avalc='INDETERMINATE' then
      avalc='IND';

   if avalc='NEGATIVE' then
      avalc='NEG';
run;

** Create status values results. **;

proc sort data=adsympt;
   by usubjid avisitn paramn aval avalc adt astdt;
run;

data symp_all_1 ord_data_1(keep=usubjid visitnum visit avisitn avisit srtdt)
      vis_colsp1(keep=usubjid visitnum visit avisitn avisit);
   recseq=put(_n_, z7.);
   set adsympt(keep=usubjid paramn paramcd param parcat1 aval avalc visitnum
      visit avisitn avisit adt astdt aendt);
   by usubjid avisitn paramn aval avalc adt astdt;
   stat=input(put(avalc, $stat.), ?? best.);
   srtdt=astdt;
   ** Group Symptoms and test results. **;

   if strip(paramcd) in ('CHILLS', 'DIARRHEA', 'FEVER', 'NLTSTSML', 'NCOUG',
      'NSTBRTH', 'NMUSPN', 'NSRTHROT', 'VOMIT') then
         do;
         grp=1;
         output symp_all_1;
      end;

   if strip(paramcd) in ('CHILLS', 'DIARRHEA', 'FEVER', 'NLTSTSML', 'NCOUG',
```

```
      'NSTBRTH', 'NMUSPN', 'NSRTHROT', 'VOMIT') or strip(paramcd) in ('FATIGUE',
      'HEADACHE', 'RIHNRA', 'NAUSEA', 'NNSLCONG') then
         do;
         grp=2;
         output symp_all_1;
      end;

   if strip(paramcd) in ('C19NIG') and strip(avisit) in ('V1_DAY1_VAX1_L',
      'V401_DAY1_VAX1') then
         do;
         grp=21;
         ** These number assignments are used below. **;
         srtdt=adt;
         output symp_all_1;
      end;

   if strip(paramcd) in ('RTCOV2NS') then
      do;
         grp=22;
         srtdt=adt;
         output symp_all_1;
      end;

   if strip(paramcd) in ('SARSCOV2') then
      do;
         grp=23;
         srtdt=adt;
         output symp_all_1;
      end;

   if strip(paramcd) in ('C19NIG') and strip(avisit) not in ('V1_DAY1_VAX1_L',
      'V401_DAY1_VAX1') then
         do;
         grp=24;
         srtdt=adt;
         output symp_all_1;
      end;

   if grp ^=. then
      output ord_data_1;

   if visitnum ^=avisitn or visit ^=avisit then
      output vis_colsp1;
   format adt astdt aendt srtdt yymmdd10.;
run;

proc sort data=ord_data_1 out=ord_data_1a noduprecs;
   by usubjid srtdt avisitn avisit visitnum visit;
   where srtdt ^=. and avisit not in('V1_DAY1_VAX1_L', 'V2_VAX2_L',
      'V401_DAY1_VAX1', 'V402_VAX2');
run;

data ord_data_1b;
   set ord_data_1a;
```

```
    by usubjid srtdt avisitn avisit visitnum visit;
    length avislist $1000;

    if first.usubjid then
       do;
          avislist='';
          srtord=10;
       end;
    id=prxparse('/' || strip(avisit) || '/i');
    call prxsubstr(id, avislist, point, lng);

    if first.usubjid or (first.avisitn and lng=0) then
       do;
          srtord + 2;
          keepflg=1;
          avislist=strip(strip(avislist) || ' ' || strip(avisit));
       end;

    if last.usubjid then
       lastrec=1;
    retain avislist;
run;

proc sql;
    create table ord_data_1c as select * from (select distinct * from ord_data_1)
       left join (select srtord from ord_data_1b as b where keepflg=1) on
       usubjid=b.usubjid and avisitn=b.avisitn and avisit=b.avisit order by usubjid,
       avisitn, srtord, srtdt;
quit;

data ord_data_1d;
    set ord_data_1c;
    by usubjid avisitn srtord srtdt;

    if first.usubjid then
       srtord_b=0;

    if avisit in ('V1_DAY1_VAX1_L', 'V2_VAX2_L', 'V401_DAY1_VAX1', 'V402_VAX2')
       then
          do;

          if strip(avisit) in ('V1_DAY1_VAX1_L', 'V401_DAY1_VAX1') then
             do;
                srtord_b=srtord_b + 1;
                srtord=srtord_b;
             end;

          if strip(avisit) in ('V2_VAX2_L', 'V402_VAX2') then
             do;
                srtord_b=srtord_b + 1;
                srtord=srtord_b;
             end;
       end;
    else
```

```
      do;

         if srtord=. then
            do;

               if first.usubjid then
                  srtord=10.1;
               else
                  srtord=srtord_ + .1;
            end;
         srtord_=srtord;
      end;
   retain srtord_b srtord_;
run;

proc sort data=vis_colsp1 out=vis_colsp2 nodupkey;
   by usubjid avisitn avisit;
run;

proc sql;
   ** Merge sort order. **;
   create table symp_all_2 as select * from symp_all_1 left join (select srtord
      from ord_data_1d as b) on usubjid=b.usubjid and visitnum=b.visitnum and
      visit=b.visit and avisitn=b.avisitn and avisit=b.avisit and srtdt=b.srtdt;
   ** Flag collapsed visits records. **;
   create table symp_all_3 as select * from symp_all_2 left join (select 1 as
      clspflg, avisitn as avisitn_colsp, avisit as avisit_colsp from vis_colsp2 as
      b) on usubjid=b.usubjid and avisitn=b.avisitn and avisit=b.avisit;
   ** Merge Death date. **;
   create table symp_all_4 as select * from symp_all_3 left join (select dthdt,
      vax101dt, vax102dt from adsl as b) on usubjid=b.usubjid order by usubjid,
      avisitn, avisit, grp, stat, astdt, visitnum, aendt;
quit;

data symp_all_5 symp1(keep=recseq usubjid vax101dt vax102dt avisitn avisit
      parcat1 grp stat dthdt srtord clspflg grpcat grp_stdt grp_endt visitnum_
      visit_ rename=(grp_stdt=astdt grp_endt=aendt visitnum_=visitnum
      visit_=visit)) nva_naat1(keep=recseq usubjid vax101dt vax102dt visitnum visit
      avisitn avisit paramn paramcd param parcat1 aval avalc grp adt srtord dthdt
      stat clspflg);
   set symp_all_4;
   by usubjid avisitn avisit grp stat astdt visitnum aendt;

   if avisitn_colsp ^=. then
      clspflg=1;

   if grp in (1, 2) then
      grpcat=1;

   if first.grp then
      do;
         grp_stdt=astdt;
         grp_endt=aendt;
         grp_stat=stat;
```

```
        visitnum_=visitnum;
        visit_=visit;
      end;

   if grp_stat < stat or grp in (7, 8) then
      do;
        grp_stdt=astdt;
        grp_endt=aendt;
        grp_stat=stat;
        visitnum_=visitnum;
        visit_=visit;
      end;

   if grp_stdt=. and stat=4 then
      grp_stdt=astdt;

   if (. < grp_endt < aendt) or aendt=. then
      grp_endt=aendt;

   if last.grp and grp < 20 then
      keepflg=1;
   output symp_all_5;

   if keepflg=1 then
      output symp1;

   if grp in (21, 22, 23, 24) then
      output nva_naat1;
   format grp_stdt grp_endt yymmdd10.;
   retain grp_stdt grp_endt grp_stat visitnum_ visit_;
run;

proc sort data=symp1;
   by usubjid avisitn avisit grpcat grp stat astdt aendt visitnum;
run;

data symp2(drop=vis_endtfl setflg vis_stat vis_astdt vis_aendt vis_endtfl_cdc
      setflg_cdc vis_stat_cdc vis_astdt_cdc vis_aendt_cdc);
   set symp1;
   by usubjid avisitn avisit grpcat grp stat astdt aendt visitnum;

   if first.avisitn then
      do;
        vis_endtfl=0;
        vis_endtfl_cdc=0;
        setflg=0;
        setflg_cdc=0;
      end;

   if setflg=0 and 3 <=grp <=7 then
      do;
        vis_stat=stat;
        vis_astdt=astdt;
        vis_aendt=aendt;
```

```
      setflg=1;
   end;

if setflg_cdc=0 and 8 <=grp <=9 then
   do;
      vis_stat_cdc=stat;
      vis_astdt_cdc=astdt;
      vis_aendt_cdc=aendt;
      setflg_cdc=1;
   end;

if 3 <=grp <=7 then
   do;

      if aendt=. or stat ^=4 then
         vis_endtfl=1;

      if vis_stat <=stat then
         do;
            vis_stat=stat;

            if vis_astdt=. or (vis_astdt ^=. and . < astdt < vis_astdt) then
               vis_astdt=astdt;
         end;

      if vis_stat=stat and astdt < vis_astdt then
         vis_astdt=astdt;

      if . < vis_aendt < aendt then
         vis_aendt=aendt;
   end;

if 8 <=grp <=9 then
   do;

      if aendt=. or stat ^=4 then
         vis_endtfl_cdc=1;

      if vis_stat_cdc <=stat then
         do;
            vis_stat_cdc=stat;

            if vis_astdt_cdc=. or (vis_astdt_cdc ^=. and . < astdt < vis_astdt_cdc)
               then
                  vis_astdt_cdc=astdt;
         end;

      if vis_stat_cdc=stat and astdt < vis_astdt_cdc then
         vis_astdt_cdc=astdt;

      if . < vis_aendt_cdc < aendt then
         vis_aendt_cdc=aendt;
   end;
output;
```

```
   if last.grpcat then
      do;

         if grpcat=2 then
            do;
               grp=20.1;
               stat=vis_stat;
               astdt=vis_astdt;

               if vis_endtfl=0 then
                  aendt=vis_aendt;
               else
                  aendt=.;
               parcat1='SEVERE COVID-19 SYMPTOMS';
               output;
            end;

         if grpcat=3 then
            do;
               grp=20.2;
               stat=vis_stat_cdc;
               astdt=vis_astdt_cdc;

               if vis_endtfl_cdc=0 then
                  aendt=vis_aendt_cdc;
               else
                  aendt=.;
               parcat1='SEVERE COVID-19 SYMPTOMS';
               output;
            end;
      end;
   retain vis_endtfl vis_endtfl_cdc setflg setflg_cdc vis_stat vis_astdt
      vis_aendt vis_stat_cdc vis_astdt_cdc vis_aendt_cdc;
run;

proc sql;
   ** Merge symptom dates based on VISITNUM. **;
   create table nva_naat1a as select * from nva_naat1 left join (select usubjid
      as usubjid_v, min(astdt) as astdt_sym_v format yymmdd10., max(aendt) as
      aendt_sym_v format yymmdd10. from symp_all_3 as b where grp in (1, 2) and
      astdt ^=. group by usubjid, visitnum, visit) on usubjid=b.usubjid and
      visitnum=b.visitnum and visit=b.visit order by usubjid, vax101dt, vax102dt,
      avisitn, avisit, visitnum, visit, grp, adt;
   ** Merge symptom dates based on AVISITN. **;
   create table nva_naat1b as select * from nva_naat1a left join (select usubjid
      as usubjid_av, min(astdt) as astdt_sym_av format yymmdd10., max(aendt) as
      aendt_sym_av format yymmdd10. from symp1 as b where grp in (1, 2) and
      astdt ^=. group by usubjid, avisitn, avisit) on usubjid=b.usubjid and
      avisitn=b.avisitn and avisit=b.avisit order by usubjid, vax101dt, vax102dt,
      avisitn, avisit, visitnum, visit, grp, stat, adt;
quit;

** Determine if NVA or NAAT result/s are valid based on dates to exclude multiple records that are out of window. **;
```

```
data nva_naat2 nva_naat_flags(keep=usubjid vax101dt vax102dt dthdt vldrslfl
    vrblngfl crd1ngfl crd2ngfl pdp17fl_ pdp27fl_);
  **** Use this dataset for flags *****;
  set nva_naat1b;
  by usubjid vax101dt vax102dt avisitn avisit visitnum visit grp stat adt;
  ** Derive result flags. **;

  if first.usubjid then
    do;
      vrblngfl='U';
      crd1ngfl='U';
      crd2ngfl='U';
      pdp17fl_='N';
      pdp27fl_='N';
    end;
  vldrslfl='N';

  if strip(avisit) in ('V1_DAY1_VAX1_L', 'V401_DAY1_VAX1') then
    do;

      if . < adt <=vax101dt then
        vldrslfl='Y';

      if vldrslfl='Y' and strip(put(stat, stat.))='POS' then
        do;

          if grp=21 then
            vrblngfl='N';

          if grp=22 then
            crd1ngfl='N';
        end;

      if vldrslfl='Y' and strip(put(stat, stat.))='NEG' then
        do;

          if grp=21 then
            vrblngfl='Y';

          if grp=22 then
            crd1ngfl='Y';
        end;

      if last.avisitn and vrblngfl='Y' and crd1ngfl='Y' then
        pdp17fl_='Y';
    end;
  else if strip(avisit) in ('V2_VAX2_L', 'V402_VAX2') then
    do;

      if . < adt <=vax102dt then
        vldrslfl='Y';

      if vldrslfl='Y' and strip(put(stat, stat.))='POS' and grp=22 then
```

```
            crd2ngfl='N';

         if vldrslfl='Y' and strip(put(stat, stat.))='NEG' and grp=22 then
            crd2ngfl='Y';

         if last.avisitn and vrblngfl='Y' and crd1ngfl='Y' and crd2ngfl='Y' then
            pdp27fl_='Y';
      end;
   else if strip(avisit) not in ('V1_DAY1_VAX1_L', 'V2_VAX2_L', 'V401_DAY1_VAX1',
      'V402_VAX2') and grp ^=24 then
         do;

         if usubjid_av ^='' then
            do;

               if astdt_sym_av ^=. and aendt_sym_av=. and sum(astdt_sym_av, -4) <=adt then
                  vldrslfl='Y';

               if astdt_sym_av ^=. and aendt_sym_av ^=. and sum(astdt_sym_av, -4)
                  <=adt <=sum(aendt_sym_av, 4) then
                     vldrslfl='Y';
            end;
         else if usubjid_v ^='' then
            do;

               if astdt_sym_v ^=. and aendt_sym_v=. and sum(astdt_sym_v, -4) <=adt then
                  vldrslfl='Y';
               else if astdt_sym_v ^=. and aendt_sym_v ^=. and sum(astdt_sym_v, -4)
                  <=adt <=sum(aendt_sym_v, 4) then
                     vldrslfl='Y';
            end;
         else
            cncrslfl='Y';
      end;

   if first.grp and last.grp then
      cncrslfl='Y';
   else
      do;
         ** Check if multiple results are present and valid. **;

         if vldrslfl='Y' then
            cncrslfl='Y';
      end;
   output nva_naat2;

   if last.usubjid then
      output nva_naat_flags;
   retain vrblngfl crd1ngfl crd2ngfl pdp17fl_ pdp27fl_;
run;

proc sort data=nva_naat2 out=nva_naat3(drop=usubjid_v usubjid_av);
   by usubjid vax101dt vax102dt avisitn avisit visitnum visit grp stat adt;
   where cncrslfl='Y';
```

```
run;

***** Chek Number of Subjects with VRBLNGFL='Y' and CRD1NGFL='Y' and CRD2NGFL='Y' ******;

proc sql noprint;
   select count (distinct usubjid) into :n1 from nva_naat_flags where
      VRBLNGFL='Y' and CRD1NGFL='Y' and CRD2NGFL='Y' and usubjid in (select
      distinct usubjid from adsl where saffl='Y');
quit;

data nva_naat4 nva_naat4a(keep=usubjid adt stat rename=(adt=nva_dt stat=nva))
      nva_naat4b(keep=usubjid adt stat rename=(adt=cnt_1dt stat=cnt_1))
      nva_naat4c(keep=usubjid adt stat rename=(adt=cnt_2dt stat=cnt_2))
      nva_naat4d(keep=usubjid vax101dt vax102dt avisitn avisit visitnum visit adt
      stat srtord clspflg vldrslfl rename=(vldrslfl=c_vldrslfl adt=cnt_unp_dt
      stat=cnt_unp srtord=cnt_srtord)) nva_naat4e(keep=usubjid vax101dt vax102dt
      avisitn avisit visitnum visit adt stat srtord clspflg vldrslfl
      rename=(vldrslfl=l_vldrslfl adt=lcl_unp_dt stat=lcl_unp srtord=lcl_srtord))
      nva_naat4f(keep=recseq usubjid vax101dt vax102dt adt grp stat
      rename=(recseq=recseq_f adt=nva_v3_dt stat=nva_v3)) nva_naat4g(keep=recseq
      usubjid vax101dt vax102dt adt grp stat c19cnv_dy rename=(recseq=recseq_g
      adt=nva_cnv_dt stat=nva_cnv));
   set nva_naat3;
   by usubjid vax101dt vax102dt avisitn avisit visitnum visit grp stat adt;

   if grp=24 and strip(avisit) not in ('V3_MONTH1_POSTVAX2_L',
      'V404_MONTH1_POSTVAX2') and vax102dt ^=. then
         do;
         c19cnv_dy=adt - vax102dt + 1;
      end;

   if last.grp and grp <=23 then
      keepflg=1;

   if grp=24 then
      keepflg=1;
   output nva_naat4;

   if keepflg=1 then
      do;

         if grp=21 and strip(avisit) in ('V1_DAY1_VAX1_L', 'V401_DAY1_VAX1') then
            output nva_naat4a;

         if grp=22 and strip(avisit) in ('V1_DAY1_VAX1_L', 'V401_DAY1_VAX1') then
            output nva_naat4b;

         if grp=22 and strip(avisit) in ('V2_VAX2_L', 'V402_VAX2') then
            output nva_naat4c;

         if grp=22 and strip(avisit) not in ('V1_DAY1_VAX1_L', 'V2_VAX2_L',
            'V401_DAY1_VAX1', 'V402_VAX2') then
               output nva_naat4d;
```

```
        if grp=23 and strip(avisit) not in ('V1_DAY1_VAX1_L', 'V2_VAX2_L',
           'V401_DAY1_VAX1', 'V402_VAX2') then
              output nva_naat4e;

        if grp=24 then
           do;

              if strip(avisit) in ('V3_MONTH1_POSTVAX2_L', 'V404_MONTH1_POSTVAX2') then
                 output nva_naat4f;
              else
                 output nva_naat4g;
           end;
     end;
run;

** Process central and local lab rerults. **;

data cnt_lcl1;
   merge nva_naat4d(in=a) nva_naat4e(in=b);
   by usubjid vax101dt vax102dt avisitn avisit visitnum visit;
   ** Conclude NAAT result for unplanned visits. **;

   if c_vldrslfl='Y' then
      do;
         naat_unp=cnt_unp;
         naat_unp_dt=cnt_unp_dt;
         srtord_swab=cnt_srtord;
      end;
   else if c_vldrslfl ^='Y' and l_vldrslfl='Y' then
      do;
         naat_unp=lcl_unp;
         naat_unp_dt=lcl_unp_dt;
         srtord_swab=lcl_srtord;
      end;

   if c_vldrslfl ^='Y' and l_vldrslfl ^='Y' then
      do;

         if nmiss(cnt_unp, naat_unp) < 2 then
            stat_sort=max(cnt_unp, naat_unp);

         if nmiss(cnt_unp_dt, naat_unp_dt) < 2 then
            dt_sort=max(cnt_unp_dt, naat_unp_dt);
      end;
   else
      do;
         stat_sort=naat_unp;
         dt_sort=naat_unp_dt;
      end;

   if srtord_swab=. then
      do;

         if cnt_srtord ^=. then
```

```
            srtord_swab=cnt_srtord;

          if cnt_srtord=. and lcl_srtord ^=. then
            srtord_swab=lcl_srtord;
        end;

    if c_vldrslfl='Y' or l_vldrslfl='Y' then
      vunprfl='Y';
    format naat_unp_dt yymmdd10.;
run;

proc sort data=cnt_lcl1;
    by usubjid vax101dt vax102dt avisitn avisit vunprfl naat_unp stat_sort dt_sort;
run;

data cnt_lcl2(drop=stat_sort);
    ****** Use this dataset for conlcuded lab results *****;
    set cnt_lcl1;
    by usubjid vax101dt vax102dt avisitn avisit vunprfl naat_unp stat_sort dt_sort;

    if last.avisitn then
      keepflg=1;
    naat_rslt_flg=1;
    rename visitnum=visitnum_ visit=visit_;
run;

** Merge result flags with symptom data. **;

data symp3a(drop=keepflg);
    merge symp2(in=a) cnt_lcl2(in=b where=(keepflg=1));
    by usubjid avisitn avisit;

    if a and not b then
      mflg=1;

    if a and b then
      mflg=2;

    if not a and b then
      mflg=3;

    if mflg=3 and visitnum=. then
      do;
        visitnum=visitnum_;
        visit=visit_;
        srtord=srtord_swab;
      end;
run;

data symp3b;
    merge symp3a(in=a) nva_naat_flags(in=b drop=vax101dt vax102dt vldrslfl)
      nva_naat4a(in=c) nva_naat4b(in=d) nva_naat4c(in=d);
    by usubjid;
```

```
    if a;
    call missing(stdy1, stdy2);

    if astdt ^=. then
       do;

          if vax101dt ^=. then
             do;

                if astdt >=vax101dt then
                   stdy1=(astdt - vax101dt) + 1;
                else
                   stdy1=(astdt - vax101dt);
             end;

          if vax102dt ^=. then
             do;

                if astdt >=vax102dt then
                   stdy2=(astdt - vax102dt) + 1;
                else
                   stdy2=(astdt - vax102dt);
             end;
       end;

    if vrblngfl='' then
       vrblngfl='U';

    if crd1ngfl='' then
       crd1ngfl='U';

    if crd2ngfl='' then
       crd2ngfl='U';

    if pdp17fl_='' then
       pdp17fl_='N';

    if pdp27fl_='' then
       pdp27fl_='N';
    rename pdp17fl_=pdp17fl_tmp pdp27fl_=pdp27fl_tmp;
run;

proc sort data=symp3b out=symp3c;
    by usubjid vax101dt vax102dt srtord avisitn avisit grpcat grp astdt;
run;

data symp3d;
    retain recseq usubjid parcat1 visitnum visit avisitn avisit clspflg vax101dt
       vax102dt dthdt nva nva_dt vrblngfl cnt_1 cnt_1dt crd1ngfl cnt_2 cnt_2dt
       crd2ngfl grpcat grp stat astdt aendt stdy1 stdy2 cnt_unp cnt_unp_dt
       c_vldrslfl lcl_unp lcl_unp_dt l_vldrslfl naat_unp naat_unp_dt vunprfl
       naat_rslt_flg pdp17fl_tmp pdp27fl_tmp;
    set symp3c;
    by usubjid vax101dt vax102dt srtord avisitn avisit grpcat grp astdt;
```

```
run;

** Determine NAAT unplanned result and derive case. **;
***** Use symp4 dataset to identify subjects with symptoms and no valid NEG result *****;

data symp4(drop=naat_unp_) symp_all_flags(keep=usubjid vax101dt vax102dt dthdt
      pdsymfl_ pdsdmfl_ cdcsymfl_ sevsymfl_ sevcdcfl_ pdrmufl_ pdrmupfl_ cdcrmufl_
      cdrmupfl_ pdp1fl_ pdp17fl_ pdp27fl_ pdp214fl_ cdp1fl_ cdp17fl_ cdp27fl_
      cdp214fl_);
   set symp3d end=eof;
   by usubjid vax101dt vax102dt srtord avisitn avisit grpcat grp astdt;
   ** Setting the flags. **;

   if first.usubjid then
      do;
         pdsymfl_='N';
         pdsdmfl_='N';
         cdcsymfl_='N';
         sevsymfl_='N';
         sevcdcfl_='N';
         pdrmufl_='N';
         pdrmupfl_='N';
         cdcrmufl_='N';
         cdrmupfl_='N';
         pdp1fl_=pdp17fl_tmp;
         pdp17fl_=pdp17fl_tmp;
         pdp27fl_=pdp27fl_tmp;
         pdp214fl_=pdp27fl_tmp;
         cdp1fl_=pdp1fl_;
         cdp17fl_=pdp17fl_;
         cdp27fl_=pdp27fl_;
         cdp214fl_=pdp27fl_;
         filocrfl_pd_='';
         filocrfl_cdc_='';
         filocrfl_sev_='';
         filocrfl_sev_cdc_='';
         pd_fst_pos_dt=.;
         astdt_pd_res_miss=.;
         cd_fst_pos_dt=.;
         astdt_cdc_res_miss=.;
         last_vis_end_dt=.;
      end;
   ** If concluded lab result out of CDC defined symptoms date/s, reset the valid flag. **;

   if grp=2 then
      do;

         if astdt=. or naat_unp_dt=. then
            vunprfl='';
         else
            do;

               if aendt=. and sum(astdt, -4) <=naat_unp_dt then
                  vunprfl='Y';
```

```
            else if aendt ^=. and sum(astdt, -4) <=naat_unp_dt <=sum(aendt, 4) then
               vunprfl='Y';
            else
               vunprfl='';
         end;
   end;
** Determine illness onset for protocol defined, CDC defined and severe symptoms. **;

if first.avisitn then
   do;
      c19onst_=-1;
      cdconst_=-1;
   end;

if strip(put(stat, stat.))='POS' and vunprfl='Y' then
   do;

      if strip(put(naat_unp, stat.)) in ('') then
         naat_unp_=input(put('UNK', $stat.), ?? best.);
      else
         naat_unp_=naat_unp;

      if grp=1 then
         c19onst=naat_unp_;

      if grp=2 then
         cdconst=naat_unp_;
   end;
else if strip(put(stat, stat.))='POS' and vunprfl='' then
   do;

      if grp=1 then
         c19onst=input(put('UNK', $stat.), ?? best.);

      if grp=2 then
         cdconst=input(put('UNK', $stat.), ?? best.);
   end;
else if strip(put(stat, stat.)) in ('', 'NEG') then
   do;

      if grp=1 then
         c19onst=input(put('NEG', $stat.), ?? best.);

      if grp=2 then
         cdconst=input(put('NEG', $stat.), ?? best.);
   end;

if grp=1 then
   c19onst_=c19onst;

if grp=2 then
   cdconst_=cdconst;

if grp=20.1 then
```

```
   do;

      if c19onst_=-1 then
         c19onst_=2;

      if strip(put(stat, stat.))='POS' then
         sevconst=c19onst_;

      if strip(put(stat, stat.)) in ('', 'NEG') then
         sevconst=input(put('NEG', $stat.), ?? best.);

      if strip(put(c19onst_, stat.))='POS' and last.usubjid and dthdt ^=. then
         sevconst=input(put('POS', $stat.), ?? best.);
   end;

if grp=20.2 then
   do;

      if cdconst_=-1 then
         cdconst_=2;

      if strip(put(stat, stat.))='POS' then
         cdcsonst=cdconst_;

      if strip(put(stat, stat.)) in ('', 'NEG') then
         cdcsonst=input(put('NEG', $stat.), ?? best.);

      if strip(put(cdconst_, stat.))='POS' and last.usubjid and dthdt ^=. then
         cdcsonst=input(put('POS', $stat.), ?? best.);
   end;

if strip(put(c19onst, stat.))='POS' and pd_fst_pos_dt=. then
   pd_fst_pos_dt=astdt;

if strip(put(cdconst, stat.))='POS' and cd_fst_pos_dt=. then
   cd_fst_pos_dt=astdt;

if grp=1 then
   do;

      if strip(put(stat, stat.))='POS' then
         do;
            pdsymfl_='Y';

            if astdt=. then
               pdsdmfl_='Y';

            if strip(put(c19onst, stat.)) not in ('NEG', 'POS') then
               do;
                  astdt_pd_res_miss=astdt;

                  if (pd_fst_pos_dt=.) or (. < astdt < pd_fst_pos_dt) then
                     do;
                        pdrmufl_='Y';
```

```
                    pdrmupfl_='Y';
                end;
            end;

            if strip(put(c19onst, stat.))='POS' and pdrmupfl_='Y'
               and . < astdt_pd_res_miss < pd_fst_pos_dt then
                  pdrmupfl_='N';
         end;
   end;

if grp=2 then
   do;

      if strip(put(stat, stat.))='POS' then
         do;
            cdcsymfl_='Y';

            if strip(put(cdconst, stat.)) not in ('NEG', 'POS') then
               do;
                  astdt_cdc_res_miss=astdt;

                  if (cd_fst_pos_dt=.) or (. < astdt < cd_fst_pos_dt) then
                     do;
                        cdcrmufl_='Y';
                        cdrmupfl_='Y';
                     end;
               end;

            if strip(put(cdconst, stat.))='POS' and cdrmupfl_='Y'
               and . < astdt_cdc_res_miss < cd_fst_pos_dt then
                  cdrmupfl_='N';
         end;
   end;

if grp=20.1 and strip(put(stat, stat.))='POS' then
   sevsymfl_='Y';

if grp=20.2 and strip(put(stat, stat.))='POS' then
   sevcdcfl_='Y';

if dthdt ^=. then
   do;
      sevsymfl_='Y';
      sevcdcfl_='Y';
   end;

if grp=1 and c19onst=input(put('POS', $stat.), ?? best.) then
   do;

      if (vrblngfl='Y' and crd1ngfl='Y' and . < vax101dt=astdt)
         or (. < vax101dt < astdt) then
            ild1fl_pd='Y';
      else
         ild1fl_pd='N';
```

```
      if . < vax101dt < sum(vax101dt, 7) <=astdt then
         ild17fl_pd='Y';
      else
         ild17fl_pd='N';

      if (crd2ngfl='Y' and . < vax102dt=astdt) or (. < vax102dt < astdt) then
         ild2fl_pd='Y';
      else
         ild2fl_pd='N';

      if . < vax102dt < sum(vax102dt, 7) <=astdt then
         ild27fl_pd='Y';
      else
         ild27fl_pd='N';

      if . < vax102dt < sum(vax102dt, 14) <=astdt then
         ild214fl_pd='Y';
      else
         ild214fl_pd='N';

      if filocrfl_pd_='' then
         do;
            filocrfl_pd_='Y';
            filocrfl_pd='Y';
         end;
   end;

if grp=2 and cdconst=input(put('POS', $stat.), ?? best.) then
   do;

      if (vrblngfl='Y' and crd1ngfl='Y' and . < vax101dt=astdt)
         or (. < vax101dt < astdt) then
            ild1fl_cdc='Y';
      else
         ild1fl_cdc='N';

      if . < vax101dt < sum(vax101dt, 7) <=astdt then
         ild17fl_cdc='Y';
      else
         ild17fl_cdc='N';

      if (crd2ngfl='Y' and . < vax102dt=astdt) or (. < vax102dt < astdt) then
         ild2fl_cdc='Y';
      else
         ild2fl_cdc='N';

      if . < vax102dt < sum(vax102dt, 7) <=astdt then
         ild27fl_cdc='Y';
      else
         ild27fl_cdc='N';

      if . < vax102dt < sum(vax102dt, 14) <=astdt then
         ild214fl_cdc='Y';
```

```
      else
         ild214fl_cdc='N';

      if filocrfl_cdc_='' then
         do;
            filocrfl_cdc_='Y';
            filocrfl_cdc='Y';
         end;
   end;

if (strip(put(stat, stat.))=('POS') and strip(put(naat_unp, stat.)) ^='NEG')
   and stdy1 < 1 then
      do;

      if (vax101dt ^=. and naat_unp_dt ^=. and vax101dt <=naat_unp_dt) or
         naat_unp_dt=. or vunprfl='Y' then
            do;

            if grp=1 then
               pdp1fl_='N';

            if grp=2 then
               cdp1fl_='N';
         end;
   end;

if (strip(put(stat, stat.))=('POS') and strip(put(naat_unp, stat.)) ^='NEG')
   and stdy1 < 8 then
      do;

      if (vax101dt ^=. and naat_unp_dt ^=. and
         vax101dt <=naat_unp_dt < sum(vax101dt, 7)) or naat_unp_dt=. or vunprfl='Y'
         then
            do;

            if grp=1 then
               do;
                  pdp17fl_='N';
                  pdp27fl_='N';
               end;

            if grp=2 then
               do;
                  cdp17fl_='N';
                  cdp27fl_='N';
               end;
         end;
   end;

if (strip(put(stat, stat.))=('POS') and strip(put(naat_unp, stat.)) ^='NEG')
   and stdy2 < 8 then
      do;

      if (vax101dt ^=. and vax102dt ^=. and naat_unp_dt ^=. and
```

```
      vax101dt <=naat_unp_dt < sum(vax102dt, 7)) or naat_unp_dt=. or vunprfl='Y'
      then
        do;

        if grp=1 then
          pdp27fl_='N';

        if grp=2 then
          cdp27fl_='N';
      end;
  end;

if (strip(put(stat, stat.))=('POS') and strip(put(naat_unp, stat.)) ^='NEG')
  and stdy2 < 15 then
    do;

    if (vax101dt ^=. and vax102dt ^=. and naat_unp_dt ^=. and
      vax101dt <=naat_unp_dt < sum(vax102dt, 14)) or naat_unp_dt=. or vunprfl='Y'
      then
        do;

        if grp=1 then
          pdp214fl_='N';

        if grp=2 then
          cdp214fl_='N';
      end;
  end;

if strip(put(naat_unp, stat.))='POS' and vunprfl='Y' then
  do;

    if . < naat_unp_dt < vax101dt then
      do;
        pdp1fl_='N';
      end;

    if vax101dt ^=. and vax101dt <=naat_unp_dt < sum(vax101dt, 7) then
      do;
        pdp17fl_='N';
        pdp27fl_='N';
        cdp17fl_='N';
        cdp27fl_='N';
      end;

    if vax101dt ^=. and vax102dt ^=. and vax101dt <=naat_unp_dt < sum(vax102dt,
      7) then
        do;
        pdp27fl_='N';
        cdp27fl_='N';
      end;

    if vax101dt ^=. and vax102dt ^=. and vax101dt <=naat_unp_dt < sum(vax102dt,
      14) then
```

```
            do;
            pdp214fl_='N';
            cdp214fl_='N';
         end;
   end;

if strip(put(stat, stat.)) ^='POS' and ((strip(put(cnt_unp, stat.))='POS')
   or (strip(put(cnt_unp, stat.))='' and strip(put(lcl_unp, stat.))='POS')) then
      do;

      if strip(put(cnt_unp, stat.))='POS' then
         do;
            tmp_unp=cnt_unp;
            tmp_unp_dt=cnt_unp_dt;
         end;
      else if strip(put(lcl_unp, stat.))='POS' then
         do;
            tmp_unp=lcl_unp;
            tmp_unp_dt=lcl_unp_dt;
         end;

      if . < tmp_unp_dt < vax101dt then
         do;
            pdp1fl_='N';
         end;

      if vax101dt ^=. and vax101dt <=tmp_unp_dt < sum(vax101dt, 7) then
         do;
            pdp17fl_='N';
            pdp27fl_='N';
            cdp17fl_='N';
            cdp27fl_='N';
         end;

      if vax101dt ^=. and vax102dt ^=. and vax101dt <=tmp_unp_dt < sum(vax102dt,
         7) then
            do;
            pdp27fl_='N';
            cdp27fl_='N';
         end;

      if vax101dt ^=. and vax102dt ^=. and vax101dt <=tmp_unp_dt < sum(vax102dt,
         14) then
            do;
            pdp214fl_='N';
            cdp214fl_='N';
         end;
   end;

if aendt ^=. then
   last_vis_end_dt=aendt;
output symp4;

if last.usubjid then
```

```
      output symp_all_flags;
   format naat_unp_dt last_vis_end_dt yymmdd10.;
   retain c19onst_ cdconst_ pdsymfl_ pdsdmfl_ cdcsymfl_ sevsymfl_ sevcdcfl_
      pdrmufl_ pdrmupfl_ cdcrmufl_ cdrmupfl_ filocrfl_pd_ filocrfl_cdc_
      filocrfl_sev_ filocrfl_sev_cdc_ pdp1fl_ pdp17fl_ pdp27fl_ pdp214fl_ cdp1fl_
      cdp17fl_ cdp27fl_ cdp214fl_ pd_fst_pos_dt cd_fst_pos_dt astdt_pd_res_miss
      astdt_cdc_res_miss last_vis_end_dt;
run;

/**** END - Setting up ADC19EF dataset *****/;
****** Identify subjects with Visit 3 (V3_MONTH1_POSTVAX2_L), Convalscent Visits (A1,B1,C1,...), V101 and
V201 visits from SV domains *****;
***** Subjects Without Evidence of Infection up to 1 Month After Dose 2 of BNT162b2 for 12 <= agetr01 <= 25 and
EVAL02FL = "Y" and PHASEN >= 2 ******;
****** Identify subjects with Scheduled Visit, Convalscent Visits (A1,B1,C1,...), V101 and V201 visits from SV
domains *****;

proc sql noprint;
   create table sv_v1 as select a.usubjid, input(a.svstdtc, yymmdd10.) as v3dt
      format=date9., b.subjid from dataprot.sv (where=(visit
      in ("V3_MONTH1_POSTVAX2_L") and not missing(SVSTDTC))) a inner join
      adsl (where=(phasen >=2)) b on a.usubjid=b.usubjid order by usubjid;
   create table sv_conv_1 as select a.usubjid, input(a.svstdtc, yymmdd10.) as
      convdt format=date9., b.subjid from
      dataprot.sv (where=(substr(scan(strip(visit), -1, '_'), 1, 2) in ('A1', 'B1',
      'C1', 'D1', 'E1', 'F1', 'G1', 'H1') and not missing(SVSTDTC))) a inner join
      adsl (where=(phasen >=2)) b on a.usubjid=b.usubjid order by usubjid;
   create table sv_V101_1 as select a.usubjid, input(a.svstdtc, yymmdd10.) as
      V101dt format=date9., b.subjid from dataprot.sv (where=(visit
      in ('V101_VAX3') and not missing(SVSTDTC))) a inner join
      adsl (where=(phasen >=2)) b on a.usubjid=b.usubjid order by usubjid;
   create table sv_V201_1 as select a.usubjid, input(a.svstdtc, yymmdd10.) as
      V201dt format=date9., b.subjid from dataprot.sv (where=(visit
      in ('V201_SURVEIL_CONSENT') and not missing(SVSTDTC))) a inner join
      adsl (where=(phasen >=2)) b on a.usubjid=b.usubjid order by usubjid;
   create table dt_1 as select a.usubjid, a.vax101dt, a.vax102dt, a.vax201dt,
      a.vax202dt, a.subjid, b.v3dt, c.convdt, d.V101dt, e.V201dt, case when not
      missing(a.vax102dt) and not missing(c.convdt) then c.convdt - a.vax102dt + 1
      else . end as convdy, case when not missing(a.vax102dt) and not
      missing(d.v101dt) then d.v101dt - a.vax102dt + 1 else . end as v101dy, case
      when not missing(a.vax102dt) and not missing(e.v201dt) then
      e.v201dt - a.vax102dt + 1 else . end as v201dy from adsl (where=(phasen >=2))
      a left join sv_v1 b on a.usubjid=b.usubjid left join sv_conv_1 c on
      a.usubjid=c.usubjid left join sv_V101_1 d on a.usubjid=d.usubjid left join
      sv_V201_1 e on a.usubjid=e.usubjid order by usubjid;
quit;

****** Identify subjects with C19NIG results for Scheduled Visit, Convalscent Visits (A1,B1,C1,...), V101 and V201
visits from ADSYMPT domains *****;

proc sql noprint;
   create table c19_v_1 as select a.usubjid, a.adt as c19v3dt format=date9.,
      a.avalc as c19val3, b.subjid from adsympt (where=(visit
      in ("V3_MONTH1_POSTVAX2_L") and paramcd in ('C19NIG'))) a inner join
```

```
      adsl (where=(phasen >=2)) b on a.usubjid=b.usubjid order by usubjid;
   create table c19_conv_1 as select a.usubjid, a.adt as c19cnvdt format=date9.,
      a.avalc as c19valc, b.subjid from adsympt (where=(substr(scan(strip(visit),
      -1, '_'), 1, 2) in ('A1', 'B1', 'C1', 'D1', 'E1', 'F1', 'G1', 'H1') and
      paramcd in ('C19NIG'))) a inner join adsl (where=(phasen >=2)) b on
      a.usubjid=b.usubjid order by usubjid;
   create table c19_v101_1 as select a.usubjid, a.adt as c19v101dt format=date9.,
      a.avalc as c19val11, b.subjid from adsympt (where=(visit in ('V101_VAX3') and
      paramcd in ('C19NIG'))) a inner join adsl (where=(phasen >=2)) b on
      a.usubjid=b.usubjid order by usubjid;
   create table c19_v201_1 as select a.usubjid, a.adt as c19v201dt format=date9.,
      a.avalc as c19val21, b.subjid from adsympt (where=(visit
      in ('V201_SURVEIL_CONSENT') and paramcd in ('C19NIG'))) a inner join
      adsl (where=(phasen >=2)) b on a.usubjid=b.usubjid order by usubjid;
quit;

data c19_1;
   merge c19_v_1 (in=a) c19_conv_1 (in=b) c19_v101_1 (in=c) c19_v201_1 (in=d);
   by usubjid;

   if a or b or c or d;
run;

****** Identify subjects with RTCOV2NS results for Scheduled Visit, Convalscent Visits (A1,B1,C1,...) and V201
visits from ADSYMPT domains *****;

proc sql noprint;
   create table rt_v_1 as select a.usubjid, a.adt as rtv3dt format=date9.,
      a.avalc as rtval3, b.subjid from adsympt (where=(visit
      in ("V3_MONTH1_POSTVAX2_L") and paramcd in ('RTCOV2NS'))) a inner join
      adsl (where=(phasen >=2)) b on a.usubjid=b.usubjid order by usubjid;
   create table rt_conv_1 as select a.usubjid, a.adt as rtcnvdt format=date9.,
      a.avalc as rtvalc, b.subjid from adsympt (where=(substr(scan(strip(visit),
      -1, '_'), 1, 2) in ('A1', 'B1', 'C1', 'D1', 'E1', 'F1', 'G1', 'H1') and
      paramcd in ('RTCOV2NS'))) a inner join adsl (where=(phasen >=2)) b on
      a.usubjid=b.usubjid order by usubjid;
   create table rt_v101_1 as select a.usubjid, a.adt as rtv101dt format=date9.,
      a.avalc as rtval11, b.subjid from adsympt (where=(visit in ('V101_VAX3') and
      paramcd in ('RTCOV2NS'))) a inner join adsl (where=(phasen >=2)) b on
      a.usubjid=b.usubjid order by usubjid;
   create table rt_v201_1 as select a.usubjid, a.adt as rtv201dt format=date9.,
      a.avalc as rtval21, b.subjid from adsympt (where=(visit
      in ('V201_SURVEIL_CONSENT') and paramcd in ('RTCOV2NS'))) a inner join
      adsl (where=(phasen >=2)) b on a.usubjid=b.usubjid order by usubjid;
quit;

data rt_1;
   merge rt_v_1 (in=a) rt_conv_1 (in=b) rt_v101_1 (in=c) rt_v201_1 (in=d);
   by usubjid;

   if a or b or c or d;
run;

****** With Results - Get Visit 3 Date cut off ******;
```

```
data dt_c19_rt_1 miss_vis3dt_1 nomiss_vis3dt_1;
   merge dt_1 (in=a) c19_1 rt_1;
   by usubjid;

   if a;
   format vis3dt date9.;

   if (not missing(c19val3) or not missing(rtval3)) and not missing(v3dt) then
      vis3dt=v3dt;
   else
      do;

         if (not missing(c19valc) or not missing(rtvalc)) and not missing(convdt) then
            do;

               if 28 <=convdy <=42 or (not missing(v3dt) and
                  v3dt - 7 <=convdt <=v3dt + 7) then
                     vis3dt=convdt;
            end;

         if (not missing(c19val11) or not missing(rtval11)) and not missing(v101dt)
            then
               do;

               if 28 <=v101dy <=42 or (not missing(v3dt) and
                  v3dt - 7 <=v101dt <=v3dt + 7) then
                     vis3dt=v101dt;
            end;

         if (not missing(c19val21) or not missing(rtval21)) and not missing(v201dt)
            then
               do;

               if 28 <=v201dy <=42 or (not missing(v3dt) and
                  v3dt - 7 <=v201dt <=v3dt + 7) then
                     vis3dt=v201dt;
            end;
      end;
   output dt_c19_rt_1;

   if missing(vis3dt) then
      output miss_vis3dt_1;

   if not missing(vis3dt) then
      output nomiss_vis3dt_1;
run;

proc sort data=nomiss_vis3dt_1;
   by usubjid vis3dt;
run;

data nomiss_vis3dt_1;
   set nomiss_vis3dt_1;
```

```
   by usubjid vis3dt;

   if first.usubjid;
run;

****** Combine VRBLNGFL, CRD1NGFL, CRD2NGFL and VIS3DT to check unique subjects ******;

proc sql noprint;
   create table nva_naat_vis3_1 as select a.usubjid, a.vrblngfl, a.crd1ngfl,
      a.crd2ngfl, b.vis3dt, case when a.VRBLNGFL='Y' and a.CRD1NGFL='Y' and
      a.CRD2NGFL='Y' and not missing(b.vis3dt) then "Y" else "N" end as EV1MD2FL
      label="Subject without Evidence 1MPD2" length=1 from nva_naat_flags a left
      join nomiss_vis3dt_1 b on a.usubjid=b.usubjid order by usubjid;
quit;

proc sql noprint;
   select count (distinct usubjid) into :n1 from nva_naat_vis3_1 where
      EV1MD2FL='Y' and usubjid in (select distinct usubjid from adsl where
      phasen >=2);
quit;

****** Subjects with POS swabs after concluded results ******;

proc sql noprint;
   create table pos_s_v_1 as select distinct a.usubjid, a.naat_unp,
      a.naat_unp_dt, b.vis3dt, b.vax101dt, b.vax102dt, b.vax201dt, b.vax202dt from
      cnt_lcl2 (where=(naat_unp ne 2 and keepflg=1)) a left join nomiss_vis3dt_1 b
      on a.usubjid=b.usubjid where . < b.vax101dt <=a.naat_unp_dt <=b.vis3dt order
      by usubjid, naat_unp_dt;
quit;

proc sort nodupkey;
   by usubjid;
run;

****** Subjects with POS swabs for Central Lab ******;

proc sql noprint;
   create table pos_s_c_v_1 as select distinct a.usubjid, a.cnt_unp,
      a.cnt_unp_dt, b.vis3dt, b.vax101dt, b.vax102dt, b.vax201dt, b.vax202dt from
      cnt_lcl2 (where=(cnt_unp ne 2)) a left join nomiss_vis3dt_1 b on
      a.usubjid=b.usubjid where . < b.vax101dt <=a.cnt_unp_dt <=b.vis3dt order by
      usubjid, cnt_unp_dt;
quit;

proc sort nodupkey;
   by usubjid;
run;

****** Subjects with POS swabs for Central Lab on VAX2 visit ******;

proc sql noprint;
   create table pos_s_c_v2_1 as select distinct a.usubjid, a.cnt_2, a.cnt_2dt,
      b.vis3dt, b.vax101dt, b.vax102dt, b.vax201dt, b.vax202dt from
```

```
      nva_naat4c (where=(cnt_2 ne 2)) a left join nomiss_vis3dt_1 b on
      a.usubjid=b.usubjid where . < b.vax101dt <=a.cnt_2dt <=b.vis3dt order by
      usubjid, cnt_2dt;
quit;

proc sort nodupkey;
   by usubjid;
run;

****** Subjects with Symptoms but no valid NEG result ******;

proc sql noprint;
   create table sym_n_v_1 as select distinct a.usubjid, a.astdt, a.stat,
      a.naat_unp, a.naat_unp_dt, b.vis3dt, b.vax101dt, b.vax102dt, b.vax201dt,
      b.vax202dt from symp4 (where=((not missing(astdt) or missing(astdt)) and
      stat=4 and naat_unp ne 2)) a left join nomiss_vis3dt_1 b on
      a.usubjid=b.usubjid where . < b.vax101dt <=a.astdt <=b.vis3dt order by
      usubjid, naat_unp_dt;
quit;

proc sort nodupkey;
   by usubjid;
run;

****** Subjects with missing Symptoms Start Date ******;

proc sql noprint;
   create table sym_no_date_1 as select distinct a.usubjid from
      symp4 (where=(stat=4)) a left join nomiss_vis3dt_1 b on a.usubjid=b.usubjid
      where missing(a.astdt) order by usubjid;
quit;

proc sort nodupkey;
   by usubjid;
run;

****** Subjects with POS N-binding Assay ******;

proc sql noprint;
   create table pos_n_v_1 as select distinct a.usubjid, a.avalc, a.adt, a.avisit,
      b.vis3dt, b.vax101dt, b.vax102dt, b.vax201dt, b.vax202dt from
      adsympt (where=(paramcd in ('C19NIG') and avisit not in ('V1_DAY1_VAX1_L')
      and avalc not in ('NEG'))) a left join nomiss_vis3dt_1 b on
      a.usubjid=b.usubjid where . < b.vax101dt <=a.adt <=b.vis3dt + 7 order by
      usubjid, adt;
quit;

proc sort nodupkey;
   by usubjid;
run;

****** Derive the final flag ******;

proc sort data=adsl out=_pop1 (keep=usubjid UNBLNDDT) nodupkey;
```

```
   by usubjid;
   where phasen >=2;
run;

data flg_1;
   merge _pop1 (in=a) nva_naat_vis3_1 (in=b) pos_s_c_v_1 (in=c) sym_n_v_1 (in=d)
      pos_n_v_1 (in=e) pos_s_v_1 (in=f) pos_s_c_v2_1 (in=g) sym_no_date_1 (in=h);
   by usubjid;

   if a;

   if (b and c) or (b and d) or (b and e) or (b and f) or (b and g) or (b and h)
      then
         EV1MD2FL='N';

   if missing(EV1MD2FL) then
      EV1MD2FL='N';
run;

proc sort out=_flg_1 (keep=usubjid EV1MD2FL);
   by usubjid EV1MD2FL;
run;

proc freq data=_flg_1;
   table EV1MD2FL / list;
run;

;
***** Subjects Without evidence of infection upto 1 month after 3rd dose of BNT162b2 or BNT162b2SA ******;
****** Identify subjects with Scheduled Visit, Convalscent Visits (A1,B1,C1,...), V101 and V201 visits from SV
domains *****;

proc sql noprint;
   create table sv_v2 as select a.usubjid, input(a.svstdtc, yymmdd10.) as v3dt
      format=date9., b.subjid from dataprot.sv (where=(visit
      in ("V303_MONTH1_POSTVAX3") and not missing(SVSTDTC))) a inner join
      adsl (where=(phasen >=2)) b on a.usubjid=b.usubjid order by usubjid;
   create table sv_conv_2 as select a.usubjid, input(a.svstdtc, yymmdd10.) as
      convdt format=date9., b.subjid from
      dataprot.sv (where=(substr(scan(strip(visit), -1, '_'), 1, 2) in ('A1', 'B1',
      'C1', 'D1', 'E1', 'F1', 'G1', 'H1') and not missing(SVSTDTC))) a inner join
      adsl (where=(phasen >=2)) b on a.usubjid=b.usubjid order by usubjid;
   create table sv_V201_2 as select a.usubjid, input(a.svstdtc, yymmdd10.) as
      V201dt format=date9., b.subjid from dataprot.sv (where=(visit
      in ('V201_SURVEIL_CONSENT') and not missing(SVSTDTC))) a inner join
      adsl (where=(phasen >=2)) b on a.usubjid=b.usubjid order by usubjid;
   create table dt_2 as select a.usubjid, a.vax101dt, a.vax102dt, a.vax201dt,
      a.vax202dt, a.subjid, b.v3dt, c.convdt, e.V201dt, case when not
      missing(a.vax201dt) and not missing(c.convdt) then c.convdt - a.vax201dt + 1
      else . end as convdy, case when not missing(a.vax201dt) and not
      missing(e.v201dt) then e.v201dt - a.vax201dt + 1 else . end as v201dy from
      adsl (where=(phasen >=2)) a left join sv_v2 b on a.usubjid=b.usubjid left
      join sv_conv_2 c on a.usubjid=c.usubjid left join sv_V201_2 e on
      a.usubjid=e.usubjid order by usubjid;
```

```
quit;

****** Identify subjects with C19NIG results for Scheduled Visit, Convalscent Visits (A1,B1,C1,...), V101 and V201
visits from ADSYMPT domains *****;

proc sql noprint;
   create table c19_v_2 as select a.usubjid, a.adt as c19v3dt format=date9.,
      a.avalc as c19val3, b.subjid from adsympt (where=(visit
      in ("V303_MONTH1_POSTVAX3") and paramcd in ('C19NIG'))) a inner join
      adsl (where=(phasen >=2)) b on a.usubjid=b.usubjid order by usubjid;
   create table c19_conv_2 as select a.usubjid, a.adt as c19cnvdt format=date9.,
      a.avalc as c19valc, b.subjid from adsympt (where=(substr(scan(strip(visit),
      -1, '_'), 1, 2) in ('A1', 'B1', 'C1', 'D1', 'E1', 'F1', 'G1', 'H1') and
      paramcd in ('C19NIG'))) a inner join adsl (where=(phasen >=2)) b on
      a.usubjid=b.usubjid order by usubjid;
   create table c19_v201_2 as select a.usubjid, a.adt as c19v201dt format=date9.,
      a.avalc as c19val21, b.subjid from adsympt (where=(visit
      in ('V201_SURVEIL_CONSENT') and paramcd in ('C19NIG'))) a inner join
      adsl (where=(phasen >=2)) b on a.usubjid=b.usubjid order by usubjid;
quit;

data c19_2;
   merge c19_v_2 (in=a) c19_conv_2 (in=b) c19_v201_2 (in=d);
   by usubjid;

   if a or b or d;
run;

****** Identify subjects with RTCOV2NS results for Scheduled Visit, Convalscent Visits (A1,B1,C1,...) and V201
visits from ADSYMPT domains *****;

proc sql noprint;
   create table rt_v_2 as select a.usubjid, a.adt as rtv3dt format=date9.,
      a.avalc as rtval3, b.subjid from adsympt (where=(visit
      in ("V303_MONTH1_POSTVAX3") and paramcd in ('RTCOV2NS'))) a inner join
      adsl (where=(phasen >=2)) b on a.usubjid=b.usubjid order by usubjid;
   create table rt_conv_2 as select a.usubjid, a.adt as rtcnvdt format=date9.,
      a.avalc as rtvalc, b.subjid from adsympt (where=(substr(scan(strip(visit),
      -1, '_'), 1, 2) in ('A1', 'B1', 'C1', 'D1', 'E1', 'F1', 'G1', 'H1') and
      paramcd in ('RTCOV2NS'))) a inner join adsl (where=(phasen >=2)) b on
      a.usubjid=b.usubjid order by usubjid;
   create table rt_v201_2 as select a.usubjid, a.adt as rtv201dt format=date9.,
      a.avalc as rtval21, b.subjid from adsympt (where=(visit
      in ('V201_SURVEIL_CONSENT') and paramcd in ('RTCOV2NS'))) a inner join
      adsl (where=(phasen >=2)) b on a.usubjid=b.usubjid order by usubjid;
quit;

data rt_2;
   merge rt_v_2 (in=a) rt_conv_2 (in=b) rt_v201_2 (in=d);
   by usubjid;

   if a or b or d;
run;
```

```
****** With Results - Get Visit 3 Date cut off ******;

data dt_c19_rt_2 miss_vis3dt_2 nomiss_vis3dt_2;
   merge dt_2 (in=a) c19_2 rt_2;
   by usubjid;

   if a;
   format vis3dt date9.;

   if (not missing(c19val3) or not missing(rtval3)) and not missing(v3dt) then
      vis3dt=v3dt;
   else
      do;

         if (not missing(c19valc) or not missing(rtvalc)) and not missing(convdt) then
            do;

               if 28 <=convdy <=42 or (not missing(v3dt) and
                  v3dt - 7 <=convdt <=v3dt + 7) then
                     vis3dt=convdt;
            end;

         if (not missing(c19val21) or not missing(rtval21)) and not missing(v201dt)
            then
               do;

               if 28 <=v201dy <=42 or (not missing(v3dt) and
                  v3dt - 7 <=v201dt <=v3dt + 7) then
                     vis3dt=v201dt;
            end;
      end;
   output dt_c19_rt_2;

   if missing(vis3dt) then
      output miss_vis3dt_2;

   if not missing(vis3dt) then
      output nomiss_vis3dt_2;
run;

proc sort data=nomiss_vis3dt_2;
   by usubjid vis3dt;
run;

data nomiss_vis3dt_2;
   set nomiss_vis3dt_2;
   by usubjid vis3dt;

   if first.usubjid;
run;

****** Combine VRBLNGFL, CRD1NGFL, CRD2NGFL and VIS3DT to check unique subjects ******;

proc sql noprint;
```

```
   create table nva_naat_vis3_2 as select a.usubjid, a.vrblngfl, a.crd1ngfl,
      a.crd2ngfl, b.vis3dt from nva_naat_flags a left join nomiss_vis3dt_2 b on
      a.usubjid=b.usubjid order by usubjid;
quit;

****** Subjects with POS swabs after concluded results ******;

proc sql noprint;
   create table pos_s_v_2 as select distinct a.usubjid, a.naat_unp,
      a.naat_unp_dt, b.vis3dt, b.vax101dt, b.vax102dt, b.vax201dt, b.vax202dt from
      cnt_lcl2 (where=(naat_unp ne 2 and keepflg=1)) a left join nomiss_vis3dt_2 b
      on a.usubjid=b.usubjid where . < b.vax101dt <=a.naat_unp_dt <=b.vis3dt order
      by usubjid, naat_unp_dt;
quit;

proc sort nodupkey;
   by usubjid;
run;

****** Subjects with POS swabs for Central Lab ******;

proc sql noprint;
   create table pos_s_c_v_2 as select distinct a.usubjid, a.cnt_unp,
      a.cnt_unp_dt, b.vis3dt, b.vax101dt, b.vax102dt, b.vax201dt, b.vax202dt from
      cnt_lcl2 (where=(cnt_unp ne 2)) a left join nomiss_vis3dt_2 b on
      a.usubjid=b.usubjid where . < b.vax101dt <=a.cnt_unp_dt <=b.vis3dt order by
      usubjid, cnt_unp_dt;
quit;

proc sort nodupkey;
   by usubjid;
run;

****** Subjects with POS swabs for Central Lab on VAX2 visit ******;

proc sql noprint;
   create table pos_s_c_v2_2 as select distinct a.usubjid, a.cnt_2, a.cnt_2dt,
      b.vis3dt, b.vax101dt, b.vax102dt, b.vax201dt, b.vax202dt from
      nva_naat4c (where=(cnt_2 ne 2)) a left join nomiss_vis3dt_2 b on
      a.usubjid=b.usubjid where . < b.vax101dt <=a.cnt_2dt <=b.vis3dt order by
      usubjid, cnt_2dt;
quit;

proc sort nodupkey;
   by usubjid;
run;

****** Subjects with Symptoms but no valid NEG result ******;

proc sql noprint;
   create table sym_n_v_2 as select distinct a.usubjid, a.astdt, a.stat,
      a.naat_unp, a.naat_unp_dt, b.vis3dt, b.vax101dt, b.vax102dt, b.vax201dt,
      b.vax202dt from symp4 (where=((not missing(astdt) or missing(astdt)) and
      stat=4 and naat_unp ne 2)) a left join nomiss_vis3dt_2 b on
```

```
      a.usubjid=b.usubjid where . < b.vax101dt <=a.astdt <=b.vis3dt order by
      usubjid, naat_unp_dt;
quit;

proc sort nodupkey;
   by usubjid;
run;

****** Subjects with missing Symptoms Start Date ******;

proc sql noprint;
   create table sym_no_date_2 as select distinct a.usubjid from
      symp4 (where=(stat=4)) a left join nomiss_vis3dt_2 b on a.usubjid=b.usubjid
      where missing(a.astdt) order by usubjid;
quit;

proc sort nodupkey;
   by usubjid;
run;

****** Subjects with POS N-binding Assay ******;

proc sql noprint;
   create table pos_n_v_2 as select distinct a.usubjid, a.avalc, a.adt, a.avisit,
      b.vis3dt, b.vax101dt, b.vax102dt, b.vax201dt, b.vax202dt from
      adsympt (where=(paramcd in ('C19NIG') and avisit not in ('V1_DAY1_VAX1_L')
      and avalc not in ('NEG'))) a left join nomiss_vis3dt_2 b on
      a.usubjid=b.usubjid where . < b.vax101dt <=a.adt <=b.vis3dt + 7 order by
      usubjid, adt;
quit;

proc sort nodupkey;
   by usubjid;
run;

****** Derive the final flag ******;

proc sort data=adsl out=_pop1 (keep=usubjid UNBLNDDT) nodupkey;
   by usubjid;
   where phasen >=2;
run;

***** Subjects Without evidence of infection upto 1 month after 4th dose of BNT162b2SA ******;
****** Identify subjects with Scheduled Visit, Convalscent Visits (A1,B1,C1,...), V101 and V201 visits from SV
domains *****;

proc sql noprint;
   create table sv_v3 as select a.usubjid, input(a.svstdtc, yymmdd10.) as v3dt
      format=date9., b.subjid from dataprot.sv (where=(visit
      in ("V305_MONTH1_POSTVAX4") and not missing(SVSTDTC))) a inner join
      adsl (where=(phasen >=2)) b on a.usubjid=b.usubjid order by usubjid;
   create table sv_conv_3 as select a.usubjid, input(a.svstdtc, yymmdd10.) as
      convdt format=date9., b.subjid from
      dataprot.sv (where=(substr(scan(strip(visit), -1, '_'), 1, 2) in ('A1', 'B1',
```

```
      'C1', 'D1', 'E1', 'F1', 'G1', 'H1') and not missing(SVSTDTC))) a inner join
      adsl (where=(phasen >=2)) b on a.usubjid=b.usubjid order by usubjid;
   create table sv_V201_3 as select a.usubjid, input(a.svstdtc, yymmdd10.) as
      V201dt format=date9., b.subjid from dataprot.sv (where=(visit
      in ('V201_SURVEIL_CONSENT') and not missing(SVSTDTC))) a inner join
      adsl (where=(phasen >=2)) b on a.usubjid=b.usubjid order by usubjid;
   create table dt_3 as select a.usubjid, a.vax101dt, a.vax102dt, a.vax201dt,
      a.vax202dt, a.subjid, b.v3dt, c.convdt, e.V201dt, case when not
      missing(a.vax202dt) and not missing(c.convdt) then c.convdt - a.vax202dt + 1
      else . end as convdy, case when not missing(a.vax202dt) and not
      missing(e.v201dt) then e.v201dt - a.vax202dt + 1 else . end as v201dy from
      adsl (where=(phasen >=2)) a left join sv_v3 b on a.usubjid=b.usubjid left
      join sv_conv_3 c on a.usubjid=c.usubjid left join sv_V201_3 e on
      a.usubjid=e.usubjid order by usubjid;
quit;

****** Identify subjects with C19NIG results for Scheduled Visit, Convalscent Visits (A1,B1,C1,...), V101 and V201
visits from ADSYMPT domains *****;

proc sql noprint;
   create table c19_v_3 as select a.usubjid, a.adt as c19v3dt format=date9.,
      a.avalc as c19val3, b.subjid from adsympt (where=(visit
      in ("V305_MONTH1_POSTVAX4") and paramcd in ('C19NIG'))) a inner join
      adsl (where=(phasen >=2)) b on a.usubjid=b.usubjid order by usubjid;
   create table c19_conv_3 as select a.usubjid, a.adt as c19cnvdt format=date9.,
      a.avalc as c19valc, b.subjid from adsympt (where=(substr(scan(strip(visit),
      -1, '_'), 1, 2) in ('A1', 'B1', 'C1', 'D1', 'E1', 'F1', 'G1', 'H1') and
      paramcd in ('C19NIG'))) a inner join adsl (where=(phasen >=2)) b on
      a.usubjid=b.usubjid order by usubjid;
   create table c19_v201_3 as select a.usubjid, a.adt as c19v201dt format=date9.,
      a.avalc as c19val21, b.subjid from adsympt (where=(visit
      in ('V201_SURVEIL_CONSENT') and paramcd in ('C19NIG'))) a inner join
      adsl (where=(phasen >=2)) b on a.usubjid=b.usubjid order by usubjid;
quit;

data c19_3;
   merge c19_v_3 (in=a) c19_conv_3 (in=b) c19_v201_3 (in=d);
   by usubjid;

   if a or b or d;
run;

****** Identify subjects with RTCOV2NS results for Scheduled Visit, Convalscent Visits (A1,B1,C1,...) and V201
visits from ADSYMPT domains *****;

proc sql noprint;
   create table rt_v_3 as select a.usubjid, a.adt as rtv3dt format=date9.,
      a.avalc as rtval3, b.subjid from adsympt (where=(visit
      in ("V305_MONTH1_POSTVAX4") and paramcd in ('RTCOV2NS'))) a inner join
      adsl (where=(phasen >=2)) b on a.usubjid=b.usubjid order by usubjid;
   create table rt_conv_3 as select a.usubjid, a.adt as rtcnvdt format=date9.,
      a.avalc as rtvalc, b.subjid from adsympt (where=(substr(scan(strip(visit),
      -1, '_'), 1, 2) in ('A1', 'B1', 'C1', 'D1', 'E1', 'F1', 'G1', 'H1') and
      paramcd in ('RTCOV2NS'))) a inner join adsl (where=(phasen >=2)) b on
```

```
      a.usubjid=b.usubjid order by usubjid;
   create table rt_v201_3 as select a.usubjid, a.adt as rtv201dt format=date9.,
      a.avalc as rtval21, b.subjid from adsympt (where=(visit
      in ('V201_SURVEIL_CONSENT') and paramcd in ('RTCOV2NS'))) a inner join
      adsl (where=(phasen >=2)) b on a.usubjid=b.usubjid order by usubjid;
quit;

data rt_3;
   merge rt_v_3 (in=a) rt_conv_3 (in=b) rt_v201_3 (in=d);
   by usubjid;

   if a or b or d;
run;

****** With Results - Get Visit 3 Date cut off ******;

data dt_c19_rt_3 miss_vis3dt_3 nomiss_vis3dt_3;
   merge dt_3 (in=a) c19_3 rt_3;
   by usubjid;

   if a;
   format vis3dt date9.;

   if (not missing(c19val3) or not missing(rtval3)) and not missing(v3dt) then
      vis3dt=v3dt;
   else
      do;

         if (not missing(c19valc) or not missing(rtvalc)) and not missing(convdt) then
            do;

               if 28 <=convdy <=42 or (not missing(v3dt) and
                  v3dt - 7 <=convdt <=v3dt + 7) then
                     vis3dt=convdt;
            end;

         if (not missing(c19val21) or not missing(rtval21)) and not missing(v201dt)
            then
               do;

               if 28 <=v201dy <=42 or (not missing(v3dt) and
                  v3dt - 7 <=v201dt <=v3dt + 7) then
                     vis3dt=v201dt;
            end;
      end;
   output dt_c19_rt_3;

   if missing(vis3dt) then
      output miss_vis3dt_3;

   if not missing(vis3dt) then
      output nomiss_vis3dt_3;
run;
```

```
proc sort data=nomiss_vis3dt_3;
   by usubjid vis3dt;
run;

data nomiss_vis3dt_3;
   set nomiss_vis3dt_3;
   by usubjid vis3dt;

   if first.usubjid;
run;

****** Combine VRBLNGFL, CRD1NGFL, CRD2NGFL and VIS3DT to check unique subjects ******;

proc sql noprint;
   create table nva_naat_vis3_3 as select a.usubjid, a.vrblngfl, a.crd1ngfl,
      a.crd2ngfl, b.vis3dt from nva_naat_flags a left join nomiss_vis3dt_3 b on
      a.usubjid=b.usubjid order by usubjid;
quit;

****** Subjects with POS swabs after concluded results ******;

proc sql noprint;
   create table pos_s_v_3 as select distinct a.usubjid, a.naat_unp,
      a.naat_unp_dt, b.vis3dt, b.vax101dt, b.vax102dt, b.vax201dt, b.vax202dt from
      cnt_lcl2 (where=(naat_unp ne 2 and keepflg=1)) a left join nomiss_vis3dt_3 b
      on a.usubjid=b.usubjid where . < b.vax101dt <=a.naat_unp_dt <=b.vis3dt order
      by usubjid, naat_unp_dt;
quit;

proc sort nodupkey;
   by usubjid;
run;

****** Subjects with POS swabs for Central Lab ******;

proc sql noprint;
   create table pos_s_c_v_3 as select distinct a.usubjid, a.cnt_unp,
      a.cnt_unp_dt, b.vis3dt, b.vax101dt, b.vax102dt, b.vax201dt, b.vax202dt from
      cnt_lcl2 (where=(cnt_unp ne 2)) a left join nomiss_vis3dt_3 b on
      a.usubjid=b.usubjid where . < b.vax101dt <=a.cnt_unp_dt <=b.vis3dt order by
      usubjid, cnt_unp_dt;
quit;

proc sort nodupkey;
   by usubjid;
run;

****** Subjects with POS swabs for Central Lab on VAX2 visit ******;

proc sql noprint;
   create table pos_s_c_v2_3 as select distinct a.usubjid, a.cnt_2, a.cnt_2dt,
      b.vis3dt, b.vax101dt, b.vax102dt, b.vax201dt, b.vax202dt from
      nva_naat4c (where=(cnt_2 ne 2)) a left join nomiss_vis3dt_3 b on
      a.usubjid=b.usubjid where . < b.vax101dt <=a.cnt_2dt <=b.vis3dt order by
```

```
      usubjid, cnt_2dt;
quit;

proc sort nodupkey;
   by usubjid;
run;

****** Subjects with Symptoms but no valid NEG result ******;

proc sql noprint;
   create table sym_n_v_3 as select distinct a.usubjid, a.astdt, a.stat,
      a.naat_unp, a.naat_unp_dt, b.vis3dt, b.vax101dt, b.vax102dt, b.vax201dt,
      b.vax202dt from symp4 (where=((not missing(astdt) or missing(astdt)) and
      stat=4 and naat_unp ne 2)) a left join nomiss_vis3dt_3 b on
      a.usubjid=b.usubjid where . < b.vax101dt <=a.astdt <=b.vis3dt order by
      usubjid, naat_unp_dt;
quit;

proc sort nodupkey;
   by usubjid;
run;

****** Subjects with missing Symptoms Start Date ******;

proc sql noprint;
   create table sym_no_date_3 as select distinct a.usubjid from
      symp4 (where=(stat=4)) a left join nomiss_vis3dt_3 b on a.usubjid=b.usubjid
      where missing(a.astdt) order by usubjid;
quit;

proc sort nodupkey;
   by usubjid;
run;

****** Subjects with POS N-binding Assay ******;

proc sql noprint;
   create table pos_n_v_3 as select distinct a.usubjid, a.avalc, a.adt, a.avisit,
      b.vis3dt, b.vax101dt, b.vax102dt, b.vax201dt, b.vax202dt from
      adsympt (where=(paramcd in ('C19NIG') and avisit not in ('V1_DAY1_VAX1_L')
      and avalc not in ('NEG'))) a left join nomiss_vis3dt_3 b on
      a.usubjid=b.usubjid where . < b.vax101dt <=a.adt <=b.vis3dt + 7 order by
      usubjid, adt;
quit;

proc sort nodupkey;
   by usubjid;
run;

****** Derive the final flag ******;

proc sort data=adsl out=_pop1 (keep=usubjid UNBLNDDT) nodupkey;
   by usubjid;
   where phasen >=2;
```

```
run;

***** Subjects Without evidence of infection upto 1 month after 2nd dose of BNT162b2SA for index(upcase(trt01a),
"NAIVE") > 0 and PHASEN >= 2 ******;
****** Identify subjects with Scheduled Visit, Convalscent Visits (A1,B1,C1,...), V101 and V201 visits from SV
domains *****;

proc sql noprint;
   create table sv_v4 as select a.usubjid, input(a.svstdtc, yymmdd10.) as v3dt
      format=date9., b.subjid from dataprot.sv (where=(visit
      in ("V404_MONTH1_POSTVAX2") and not missing(SVSTDTC))) a inner join
      adsl (where=(phasen >=2)) b on a.usubjid=b.usubjid order by usubjid;
   create table sv_conv_4 as select a.usubjid, input(a.svstdtc, yymmdd10.) as
      convdt format=date9., b.subjid from
      dataprot.sv (where=(substr(scan(strip(visit), -1, '_'), 1, 2) in ('A1', 'B1',
      'C1', 'D1', 'E1', 'F1', 'G1', 'H1') and not missing(SVSTDTC))) a inner join
      adsl (where=(phasen >=2)) b on a.usubjid=b.usubjid order by usubjid;
   create table sv_V201_4 as select a.usubjid, input(a.svstdtc, yymmdd10.) as
      V201dt format=date9., b.subjid from dataprot.sv (where=(visit
      in ('V201_SURVEIL_CONSENT') and not missing(SVSTDTC))) a inner join
      adsl (where=(phasen >=2)) b on a.usubjid=b.usubjid order by usubjid;
   create table dt_4 as select a.usubjid, a.vax101dt, a.vax102dt, a.vax201dt,
      a.vax202dt, a.subjid, b.v3dt, c.convdt, e.V201dt, case when not
      missing(a.vax102dt) and not missing(c.convdt) then c.convdt - a.vax102dt + 1
      else . end as convdy, case when not missing(a.vax102dt) and not
      missing(e.v201dt) then e.v201dt - a.vax102dt + 1 else . end as v201dy from
      adsl (where=(phasen >=2)) a left join sv_v4 b on a.usubjid=b.usubjid left
      join sv_conv_4 c on a.usubjid=c.usubjid left join sv_V201_4 e on
      a.usubjid=e.usubjid order by usubjid;
quit;

****** Identify subjects with C19NIG results for Scheduled Visit, Convalscent Visits (A1,B1,C1,...), V101 and V201
visits from ADSYMPT domains *****;

proc sql noprint;
   create table c19_v_4 as select a.usubjid, a.adt as c19v3dt format=date9.,
      a.avalc as c19val3, b.subjid from adsympt (where=(visit
      in ("V404_MONTH1_POSTVAX2") and paramcd in ('C19NIG'))) a inner join
      adsl (where=(phasen >=2)) b on a.usubjid=b.usubjid order by usubjid;
   create table c19_conv_4 as select a.usubjid, a.adt as c19cnvdt format=date9.,
      a.avalc as c19valc, b.subjid from adsympt (where=(substr(scan(strip(visit),
      -1, '_'), 1, 2) in ('A1', 'B1', 'C1', 'D1', 'E1', 'F1', 'G1', 'H1') and
      paramcd in ('C19NIG'))) a inner join adsl (where=(phasen >=2)) b on
      a.usubjid=b.usubjid order by usubjid;
   create table c19_v201_4 as select a.usubjid, a.adt as c19v201dt format=date9.,
      a.avalc as c19val21, b.subjid from adsympt (where=(visit
      in ('V201_SURVEIL_CONSENT') and paramcd in ('C19NIG'))) a inner join
      adsl (where=(phasen >=2)) b on a.usubjid=b.usubjid order by usubjid;
quit;

data c19_4;
   merge c19_v_4 (in=a) c19_conv_4 (in=b) c19_v201_4 (in=d);
   by usubjid;
```

```
   if a or b or d;
run;

****** Identify subjects with RTCOV2NS results for Scheduled Visit, Convalscent Visits (A1,B1,C1,...) and V201
visits from ADSYMPT domains *****;

proc sql noprint;
   create table rt_v_4 as select a.usubjid, a.adt as rtv3dt format=date9.,
      a.avalc as rtval3, b.subjid from adsympt (where=(visit
      in ("V404_MONTH1_POSTVAX2") and paramcd in ('RTCOV2NS'))) a inner join
      adsl (where=(phasen >=2)) b on a.usubjid=b.usubjid order by usubjid;
   create table rt_conv_4 as select a.usubjid, a.adt as rtcnvdt format=date9.,
      a.avalc as rtvalc, b.subjid from adsympt (where=(substr(scan(strip(visit),
      -1, '_'), 1, 2) in ('A1', 'B1', 'C1', 'D1', 'E1', 'F1', 'G1', 'H1') and
      paramcd in ('RTCOV2NS'))) a inner join adsl (where=(phasen >=2)) b on
      a.usubjid=b.usubjid order by usubjid;
   create table rt_v201_4 as select a.usubjid, a.adt as rtv201dt format=date9.,
      a.avalc as rtval21, b.subjid from adsympt (where=(visit
      in ('V201_SURVEIL_CONSENT') and paramcd in ('RTCOV2NS'))) a inner join
      adsl (where=(phasen >=2)) b on a.usubjid=b.usubjid order by usubjid;
quit;

data rt_4;
   merge rt_v_4 (in=a) rt_conv_4 (in=b) rt_v201_4 (in=d);
   by usubjid;

   if a or b or d;
run;

****** With Results - Get Visit 3 Date cut off ******;

data dt_c19_rt_4 miss_vis3dt_4 nomiss_vis3dt_4;
   merge dt_4 (in=a) c19_4 rt_4;
   by usubjid;

   if a;
   format vis3dt date9.;

   if (not missing(c19val3) or not missing(rtval3)) and not missing(v3dt) then
      vis3dt=v3dt;
   else
      do;

         if (not missing(c19valc) or not missing(rtvalc)) and not missing(convdt) then
            do;

               if 28 <=convdy <=42 or (not missing(v3dt) and
                  v3dt - 7 <=convdt <=v3dt + 7) then
                     vis3dt=convdt;
            end;

         if (not missing(c19val21) or not missing(rtval21)) and not missing(v201dt)
            then
               do;
```

```
            if 28 <=v201dy <=42 or (not missing(v3dt) and
               v3dt - 7 <=v201dt <=v3dt + 7) then
                  vis3dt=v201dt;
         end;
      end;
   output dt_c19_rt_4;

   if missing(vis3dt) then
      output miss_vis3dt_4;

   if not missing(vis3dt) then
      output nomiss_vis3dt_4;
run;

proc sort data=nomiss_vis3dt_4;
   by usubjid vis3dt;
run;

data nomiss_vis3dt_4;
   set nomiss_vis3dt_4;
   by usubjid vis3dt;

   if first.usubjid;
run;

****** Combine VRBLNGFL, CRD1NGFL, CRD2NGFL and VIS3DT to check unique subjects ******;

proc sql noprint;
   create table nva_naat_vis3_4 as select a.usubjid, a.vrblngfl, a.crd1ngfl,
      a.crd2ngfl, b.vis3dt, case when a.VRBLNGFL='Y' and a.CRD1NGFL='Y' and
      a.CRD2NGFL='Y' and not missing(b.vis3dt) then "Y" else "N" end as EVSAD2FL
      label="Naive Subject without Evidence 1MPD2" length=1 from nva_naat_flags a
      left join nomiss_vis3dt_4 b on a.usubjid=b.usubjid order by usubjid;
quit;

****** Subjects with POS swabs after concluded results ******;

proc sql noprint;
   create table pos_s_v_4 as select distinct a.usubjid, a.naat_unp,
      a.naat_unp_dt, b.vis3dt, b.vax101dt, b.vax102dt, b.vax201dt, b.vax202dt from
      cnt_lcl2 (where=(naat_unp ne 2 and keepflg=1)) a left join nomiss_vis3dt_4 b
      on a.usubjid=b.usubjid where . < b.vax101dt <=a.naat_unp_dt <=b.vis3dt order
      by usubjid, naat_unp_dt;
quit;

proc sort nodupkey;
   by usubjid;
run;

****** Subjects with POS swabs for Central Lab ******;

proc sql noprint;
   create table pos_s_c_v_4 as select distinct a.usubjid, a.cnt_unp,
```

```
      a.cnt_unp_dt, b.vis3dt, b.vax101dt, b.vax102dt, b.vax201dt, b.vax202dt from
      cnt_lcl2 (where=(cnt_unp ne 2)) a left join nomiss_vis3dt_4 b on
      a.usubjid=b.usubjid where . < b.vax101dt <=a.cnt_unp_dt <=b.vis3dt order by
      usubjid, cnt_unp_dt;
quit;

proc sort nodupkey;
   by usubjid;
run;

****** Subjects with POS swabs for Central Lab on VAX2 visit ******;

proc sql noprint;
   create table pos_s_c_v2_4 as select distinct a.usubjid, a.cnt_2, a.cnt_2dt,
      b.vis3dt, b.vax101dt, b.vax102dt, b.vax201dt, b.vax202dt from
      nva_naat4c (where=(cnt_2 ne 2)) a left join nomiss_vis3dt_4 b on
      a.usubjid=b.usubjid where . < b.vax101dt <=a.cnt_2dt <=b.vis3dt order by
      usubjid, cnt_2dt;
quit;

proc sort nodupkey;
   by usubjid;
run;

****** Subjects with Symptoms but no valid NEG result ******;

proc sql noprint;
   create table sym_n_v_4 as select distinct a.usubjid, a.astdt, a.stat,
      a.naat_unp, a.naat_unp_dt, b.vis3dt, b.vax101dt, b.vax102dt, b.vax201dt,
      b.vax202dt from symp4 (where=((not missing(astdt) or missing(astdt)) and
      stat=4 and naat_unp ne 2)) a left join nomiss_vis3dt_4 b on
      a.usubjid=b.usubjid where . < b.vax101dt <=a.astdt <=b.vis3dt order by
      usubjid, naat_unp_dt;
quit;

proc sort nodupkey;
   by usubjid;
run;

****** Subjects with missing Symptoms Start Date ******;

proc sql noprint;
   create table sym_no_date_4 as select distinct a.usubjid from
      symp4 (where=(stat=4)) a left join nomiss_vis3dt_4 b on a.usubjid=b.usubjid
      where missing(a.astdt) order by usubjid;
quit;

proc sort nodupkey;
   by usubjid;
run;

****** Subjects with POS N-binding Assay ******;

proc sql noprint;
```

```
    create table pos_n_v_4 as select distinct a.usubjid, a.avalc, a.adt, a.avisit,
       b.vis3dt, b.vax101dt, b.vax102dt, b.vax201dt, b.vax202dt from
       adsympt (where=(paramcd in ('C19NIG') and avisit not in ('V1_DAY1_VAX1_L')
       and avalc not in ('NEG'))) a left join nomiss_vis3dt_4 b on
       a.usubjid=b.usubjid where . < b.vax101dt <=a.adt <=b.vis3dt + 7 order by
       usubjid, adt;
quit;

proc sort nodupkey;
   by usubjid;
run;

****** Derive the final flag ******;

proc sort data=adsl out=_pop1 (keep=usubjid UNBLNDDT) nodupkey;
   by usubjid;
   where phasen >=2;
run;

data flg_4;
   merge _pop1 (in=a) nva_naat_vis3_4 (in=b) pos_s_c_v_4 (in=c) sym_n_v_4 (in=d)
      pos_n_v_4 (in=e) pos_s_v_4 (in=f) pos_s_c_v2_4 (in=g) sym_no_date_4 (in=h);
   by usubjid;

   if a;

   if (b and c) or (b and d) or (b and e) or (b and f) or (b and g) or (b and h)
      then
         EVSAD2FL='N';

   if missing(EVSAD2FL) then
      EVSAD2FL='N';
run;

proc sort out=_flg_4 (keep=usubjid EVSAD2FL);
   by usubjid EVSAD2FL;
run;

****** NVA/NAAT Scheduled Visit *****;

proc sql noprint;
   create table nva_v3 as select distinct usubjid, adt as dt_v3 from adsympt
      where visit in ('V3_MONTH1_POSTVAX2_L') and upcase(paramcd) in ('C19NIG');
   create table nva_v3_conv as select distinct usubjid, convdt as dt_v3_conv from
      dt_1 where 28 <=convdy <=42;
   create table nva_v301 as select distinct usubjid, adt as dt_v301 from adsympt
      where visit in ('V301_VAX3') and upcase(paramcd) in ('C19NIG');
   create table nva_v303 as select distinct usubjid, adt as dt_v303 from adsympt
      where visit in ('V303_MONTH1_POSTVAX3') and upcase(paramcd) in ('C19NIG');
   create table nva_v305 as select distinct usubjid, adt as dt_v305 from adsympt
      where visit in ('V305_MONTH1_POSTVAX4') and upcase(paramcd) in ('C19NIG');
   create table naat_v301 as select distinct usubjid, adt as dt_tv301 from
      adsympt where visit in ('V301_VAX3') and upcase(paramcd) in ('RTCOV2NS');
   create table naat_v303 as select distinct usubjid , adt as dt_tv303 from
```

```
      adsympt where visit in ('V303_MONTH1_POSTVAX3') and upcase(paramcd)
      in ('RTCOV2NS');
quit;

proc sort data=nva_v3_conv nodupkey;
   by usubjid;
run;

proc sort data=dataprot.mh out=c19_his (keep=usubjid) nodupkey;
   by usubjid;
   where strip(upcase(mhdecod)) in ('ASYMPTOMATIC COVID-19', 'COVID-19',
      'COVID-19 PNEUMONIA', 'COVID-19 TREATMENT', 'SUSPECTED COVID-19',
      'SARS-COV-2 ANTIBODY TEST POSITIVE', 'SARS-COV-2 CARRIER',
      'SARS-COV-2 SEPSIS', 'SARS-COV-2 TEST POSITIVE', 'SARS-COV-2 VIRAEMIA',
      'MULTISYSTEM INFLAMMATORY SYNDROME IN CHILDREN');
run;

**** Merging all Flags with ADSL *****;

data adsl adsl_flg (keep=usubjid subjid EV1MD2FL dt_:);
   merge adsl (in=a) _flg_: nva_v3 (in=v3) nva_v301 (in=v301) nva_v303 (in=v303)
      nva_v305 (in=v305) naat_v301 (in=tv301) naat_v303 (in=tv303) c19_his (in=c19)
      nva_v3_conv (in=ve_conv);
   by usubjid;

   if a;

   if missing(EV1MD2FL) or phasen < 2 or trt01an not in (8, 9) then
      EV1MD2FL='N';

   if v3 or ve_conv then
      v3f=1;

   if v301 then
      v301f=1;

   if v303 then
      v303f=1;

   if v305 then
      v305f=1;

   if tv301 then
      tv301f=1;

   if tv303 then
      tv303f=1;

   if EV1MD2FL="Y" and c19 then
      EV1MD2FL="N";
run;

data adsl(drop=dt_:);
   set adsl;
```

```
   drop V3F V301F V303F V305F TV301F TV303F;
run;

*****************************************************************************************;
* Specification 9 *;
* FOLLOW UP CATEGORIES *;
* 1 - Censor Date. *;
* 2 - Follow up variables in days. *;
* 3 - Follow up categories. *;
*****************************************************************************************;
*Add BDCSRDT/X1CSRDT;

data adsl;
   set adsl;
   attrib BDCSRDT label="Double Blinded Follow-up Censor Date"
      Format=date9. X1CSRDT label="Crossover Dose1 Censor Date"
      Format=date9. STCSRDT label="Study Censor Date" Format=date9. EOSDCDT
      label="End Of Study Discontinuation Date" Format=date9.;

   if randfl="Y" then
      do;
         STCSRDT=min(eosdcdt, "&cutoff2"d);

         if (tr02sdt>. or UNBLNDDT>.) then
            do;

               if .<tr02sdt-1<"&cutoff2"d then
                  BDCSRDT=tr02sdt-1;
               else
                  BDCSRDT="&cutoff2"d;

               if .<UNBLNDDT-1<=BDCSRDT then
                  BDCSRDT=UNBLNDDT-1;

               if .<eosdcdt<"&cutoff2"d then
                  X1CSRDT=eosdcdt;
               else
                  X1CSRDT="&cutoff2"d;
            end;
         else
            do;

               if .<UNBLNDDT-1<="&cutoff2"d then
                  BDCSRDT=UNBLNDDT-1;
               else
                  BDCSRDT="&cutoff2"d;

               if .<eosdcdt<=BDCSRDT then
                  BDCSRDT=eosdcdt;
            end;
      end;
run;

*FU & categories;
```

```
data adsl;
   set adsl;
   *FUP2CUT;

   if randfl="Y" then
      do;

         if vax102dt=. then
            FUP2CUT=0;
         else if not missing(eosdcdt) then
            do;
               FUP2CUT=eosdcdt-vax102dt+1;

               if vax10udt>vax102dt>. then
                  FUP2CUT=eosdcdt-vax10udt+1;
            end;
         else
            do;
               FUP2CUT="&cutoff2"d-vax102dt+1;

               if vax10udt>vax102dt>. then
                  FUP2CUT="17Jun2021"d-vax10udt+1;
            end;

         if FUP2CUT ne . and FUP2CUT<=0 then
            FUP2CUT=0;

         if vax102dt=. then
            _FUP2CUT=0;
         else if not missing(eosdcdt) then
            do;
               _FUP2CUT=eosdcdt-vax102dt+1;
            end;
         else
            do;
               _FUP2CUT="&cutoff2"d-vax102dt+1;
            end;

         if _FUP2CUT ne . and _FUP2CUT<=0 then
            _FUP2CUT=0;
      end;
   *FUP2UNB;

   if randfl="Y" then
      do;

         if vax102dt=. then
            FUP2UNB=0;
         else if not missing(BDCSRDT) then
            do;
               FUP2UNB=BDCSRDT-vax102dt+1;

               if vax10udt>vax102dt>. then
```

```
            FUP2UNB=BDCSRDT-vax10udt+1;
        end;

      if FUP2UNB ne . and FUP2UNB<=0 then
        FUP2UNB=0;
   end;
*FPX1CUT;

if (UNBLNDDT>. or tr02sdt>.) then
   do;

      if tr02sdt=. then
        FPX1CUT=0;
      else if not missing(eosdcdt) and eosdcdt>=tr02sdt then
        FPX1CUT=eosdcdt-tr02sdt+1;
      else
        FPX1CUT="&cutoff2"d-tr02sdt+1;

      if FPX1CUT ne . and FPX1CUT<=0 then
        FPX1CUT=0;
   end;
*FUNBCUT;

if (UNBLNDDT>. or tr02sdt>.) then
   do;

      if UNBLNDDT=. then
        FUNBCUT=0;
      else if not missing(eosdcdt) then
        FUNBCUT=eosdcdt-UNBLNDDT+1;
      else
        FUNBCUT="&cutoff2"d-UNBLNDDT+1;

      if FUNBCUT ne . and FUNBCUT<=0 then
        FUNBCUT=0;
   end;
*FUP1CUT;

if randfl="Y" then
   do;

      if vax101dt=. then
        FUP1CUT=0;
      else if not missing(eosdcdt) then
        FUP1CUT=eosdcdt-vax101dt+1;
      else
        FUP1CUT="&cutoff2"d-vax101dt+1;

      if FUP1CUT ne . and FUP1CUT<=0 then
        FUP1CUT=0;
   end;
*FUP1UNB;

if randfl="Y" then
```

```
      do;

         if vax101dt=. then
            FUP1UNB=0;
         else if not missing(BDCSRDT) then
            FUP1UNB=BDCSRDT-vax101dt+1;

         if FUP1UNB ne . and FUP1UNB<=0 then
            FUP1UNB=0;
      end;
run;

%macro FUCATADD(invar=, outvar=, outvarn=);
   proc sql;
      select floor(max(&invar./28)/1)+1 into: maxloop from adsl where &invar.>.;
   quit;

   %if %length(&maxloop) %then
      %do;

         data adsl;
            set adsl;
            length &outvar. $20;

            if randfl="Y" then
               do;

                  %do x=1 %to &maxloop;

                     %if &x>1 %then
                        %do;
                           else
                           %end;

                     if %eval(1*&x-1)<=&invar./28<%eval(1*&x) then
                        do;

                           %if &x>1 %then
                              %do;
                                 &outvar.="%eval(1*&x-1)-%eval(1*&x) months";
                              %end;
                           %else
                              %do;
                                 &outvar.="%eval(1*&x-1)-%eval(1*&x) month";
                              %end;
                           &outvarn.=&x;
                        end;
                  %end;
               end;
         run;

      %end;
%mend FUCATADD;
```

```
%FUCATADD(invar=FUP2CUT, outvar=FUP2CAT1, outvarn=FUP2CA1N);
%FUCATADD(invar=FUP2UNB, outvar=FUP2CAT2, outvarn=FUP2CA2N);
%FUCATADD(invar=FPX1CUT, outvar=FPX1CAT1, outvarn=FPX1CA1N);
%FUCATADD(invar=FUP1CUT, outvar=FUP1CAT1, outvarn=FUP1CA1N);

data adsl;
   set adsl;
   label FUP2CUT="PD2 FU Time in Days: to Cutoff"
      FUP2CA1N="PD2 FU Time Cat 1 (N): to Cutoff"
      FUP2CAT1="PD2 FU Time Cat 1: to Cutoff"
      FUP2UNB="PD2 FU Time in Days: to Unblinding"
      FUP2CA2N="PD2 FU Time Cat 2 (N): to Unblinding"
      FUP2CAT2="PD2 FU Time Cat 2: to Unblinding"
      FPX1CUT="Post Xover D1 FUTM in Days: to Cutoff"
      FPX1CA1N="Post Xover D1 FUTM Cat 1 (N): to Cutoff"
      FPX1CAT1="Post Xover D1 FUTM Cat 1: to Cutoff"
      FUP1CUT="PD1 FU Time in Days: to Cutoff"
      FUP1CA1N="PD1 FU Time Cat 1 (N): to Cutoff"
      FUP1CAT1="PD1 FU Time Cat 1: to Cutoff"
      FUNBCUT="Unblinding FUTM in Days: to Cutoff"
      FUP1UNB="PD1 FU Time in Days: to Unblinding";
   new3k=DS3KFL;

   if new3k="Y" and _FUP2CUT>=168 then
      DS3KFL="Y";
   else
      DS3KFL="N";
   drop new3k _FUP2CUT;
run;

*************************************************************************************;
* Specification 10 *;
* ADD RNA & CD4 Categories *;
*************************************************************************************;

data lbrna;
   set dataprot.lb;
   where lbtestcd="HIVR_US" and lbstat ne "NOT DONE" and (index(visit,
      "V1_DAY1_VAX1") or index(visit, "V401_DAY1_VAX1")) and lbdy<=1;

   if not missing(lbstresn) then
      do;

         if lbstresn>=50 then
            do;
               RNACAT=">=50";
               RNACATN=2;
            end;
         else if lbstresn<50 then
            do;
               RNACAT="<50";
               RNACATN=1;
            end;
      end;
```

```
   else if anydigit(lborres)>0 then
      do;

         if anydigit(lborres)=1 then
            do;
               RNANUM=input(substr(lborres, 1, anyalpha(lborres)-1), best.);
            end;
         else if anydigit(lborres)>1 then
            do;

               if index(lborres, "<") then
                  do;

                     if anyalpha(lborres)>1 then
                        RNANUM=input(substr(lborres, anydigit(lborres),
                           anyalpha(lborres)-anydigit(lborres)), best.)-0.01;
                     else
                        RNANUM=input(substr(lborres, anydigit(lborres)), best.)-0.01;
                  end;
            end;

         if RNANUM>=50 then
            do;
               RNACAT=">=50";
               RNACATN=2;
            end;
         else if RNANUM<50 then
            do;
               RNACAT="<50";
               RNACATN=1;
            end;
      end;
   else
      do;

         if upcase(lborres) ne "POSITIVE" then
            do;
               RNACAT="<50";
               RNACATN=1;
            end;
      end;

proc sort;
   by usubjid;
run;

data lbcd4;
   set dataprot.lb;
   where lbtestcd="CD4" and lbstresu in ("10^9/L", "/uL") and lbstat ne
      "NOT DONE" and (index(visit, "V1_DAY1_VAX1") or index(visit,
      "V401_DAY1_VAX1")) and lbdy<=1;

   if not missing(lbstresn) then
      do;
```

```
      if 200<=lbstresn*1000<=500 then
         do;
            CD4CAT="200-500";
            CD4CATN=2;
         end;

      if .<lbstresn*1000<200 then
         do;
            CD4CAT="<200";
            CD4CATN=1;
         end;

      if 500<lbstresn*1000 then
         do;
            CD4CAT=">500";
            CD4CATN=3;
         end;
   end;
else if anydigit(lborres)>0 then
   do;

      if anydigit(lborres)=1 then
         do;
            CD4NUM=input(substr(lborres, 1, anyalpha(lborres)-2), best.);
         end;
      else if anydigit(lborres)>1 then
         do;

            if index(lborres, "<") then
               do;

                  if anyalpha(lborres)>1 then
                     CD4NUM=input(substr(lborres, anydigit(lborres),
                        anyalpha(lborres)-anydigit(lborres)), best.)-0.01;
                  else
                     CD4NUM=input(substr(lborres, anydigit(lborres)), best.)-0.01;
               end;
         end;

      if 200<=CD4NUM<=500 then
         do;
            CD4CAT="200-500";
            CD4CATN=2;
         end;

      if .<CD4NUM<200 then
         do;
            CD4CAT="<200";
            CD4CATN=1;
         end;

      if 500<CD4NUM then
         do;
```

```
            CD4CAT=">500";
            CD4CATN=3;
          end;
      end;

proc sort;
   by usubjid;
run;

data adsl;
   merge adsl(in=a) lbrna(keep=usubjid RNACAT RNACATN) lbcd4(keep=usubjid CD4CAT
      CD4CATN);
   by usubjid;

   if a;
   label RNACAT="HIV RNA Category" RNACATN="HIV RNA Category (N)"
      CD4CAT="CD4 Category for HIV-positive"
      CD4CATN="CD4 Category for HIV-positive (N)";

   if HIVFL ne "Y" then
      do;
         CD4CAT="";
         CD4CATN=.;
         RNACAT="";
         RNACATN=.;
      end;
run;

*Date of Completion;

proc sql;
   create table ds as select a.*, b.qval as dsphase from dataprot.ds a left join
      dataprot.suppds(where=(QNAM="DSPHASE" and upcase(QVAL) not in (" "))) b on
      a.usubjid=b.usubjid and a.dsseq=input(b.idvarval, best.) order by usubjid,
      dsseq;
quit;

data compdt(keep=usubjid COMPLDT);
   attrib COMPLDT label='Date of Completion' format=date9.;
   set ds;
   COMPLDT=input(DSSTDTC, e8601da.);

   if dscat='DISPOSITION EVENT';

   if dsphase='FOLLOW-UP';

   if dsdecod="COMPLETED";

proc sort;
   by usubjid;
run;

*Eot;
```

```
proc sql;
   create table ds_m as select a.*, b.qval as DSPHASE from dataprot.ds a left
      join dataprot.suppds(where=(QNAM="DSPHASE" and upcase(QVAL) not in (" "))) b
      on a.usubjid=b.usubjid and a.dsseq=input(b.idvarval, best.);
quit;

*End Of Study Discontinuation Reason;

data EOSDCRS(keep=usubjid EOSDCRS);
   set ds_m;
   attrib EOSDCRS label='End Of Study Discontinuation Reason' length=$50.;

   if dscat='DISPOSITION EVENT';

   if dsphase='FOLLOW-UP';

   if DSDECOD ^="COMPLETED";
   EOSDCRS=dsdecod;

proc sort;
   by usubjid;
run;

*End Of Treatment Discontinuation;

data eot(keep=USUBJID EOTDCDT EOTDCRS);
   attrib EOTDCRS label='End Of Treatment Discontinuation Reason' length=$50.
      EOTDCDT label='End Of Treatment Discontinuation Date' length=8. format=Date9.;
   set ds_m;
   EOTDCRS=DSDECOD;

   if not missing(DSSTDTC) then
      EOTDCDT=input(DSSTDTC, e8601da.);

   if DSCAT='DISPOSITION EVENT';

   if DSPHASE in ('VACCINATION');

   if DSDECOD ^="COMPLETED";

proc sort;
   by usubjid;
run;

proc sort data=adsl out=adsl;
   by usubjid;
run;

data adsl;
   attrib RACIALDN label='Racial Designation (N)' RACIALD
      label='Racial Designation';
   merge adsl(in=a) compdt EOSDCRS eot;
   by usubjid;
```

```
    if a;

    if RACIALD="JAPANESE" then
       RACIALDN=5;

    if RACIALD="OTHER" then
       RACIALDN=999;
run;

proc sort data=adsl out=adsl;
    by usubjid;
run;

*Add first positive date after dose 1;

proc sql noprint;
    create table cen_res as select usubjid, visit, visitnum, input(mbdtc,
       yymmdd10.) as c_adt format=date9., mbstresc as c_avalc length=5 from
       dataprot.mb where upcase(mbtestcd)="RTCOV2NS" and mbdy>1 order by usubjid,
       visitnum, visit;
    create table loc_res as select usubjid, visit, visitnum, input(mbdtc,
       yymmdd10.) as l_adt format=date9., case when spdevid in ('34', '44', '68')
       then mbstresc else "UNK" end as l_avalc length=5 from dataprot.mb where
       upcase(mbtestcd)="SARSCOV2" and mbdy>1 order by usubjid, visitnum, visit;
quit;

data pd1pos0;
    merge cen_res (in=a) loc_res (in=b);
    by usubjid visitnum visit;

    if a or b;

    if c_avalc="POS" then
       do;
          PD1POS=c_avalc;
          PD1POSDT=c_adt;
       end;

    if missing(c_avalc) and l_avalc="POS" then
       do;
          PD1POS=l_avalc;
          PD1POSDT=l_adt;
       end;

    if c_avalc="POS" and l_avalc="POS" then
       do;
          PD1POS=c_avalc;
          PD1POSDT=min(c_adt, l_adt);
       end;
    format PD1POSDT date9.;
run;

proc sort;
    by usubjid PD1POSDT;
```

```
   where not missing(PD1POSDT);
run;

data pd1pos1 (keep=usubjid PD1POSDT);
   set pd1pos0;
   by usubjid PD1POSDT;

   if first.usubjid;
   label PD1POSDT="Post-dose Positive Swab Result Date";
run;

data adsl;
   merge adsl(in=a) pd1pos1;
   by usubjid;
   if a;
run;

data adsl;
   attrib TRTDELT label="Adjusted Treatment for Rel. Efficacy" length=$100.
      TRTDELTN label="Adjusted Treatment for Rel. Efficacy (N)";
   set adsl;
   where agegr4n=1;
   if PHASEN>1 then
      do;

         if UNBLNDDT ne . and vax201dt ne . and vax202dt ne . and VAX201DT>=UNBLNDDT
            then
               do;
               TRTDELT="Placebo";
               TRTDELTN=9;
            end;
         else
            do;
               TRTDELT="BNT162b2 Phase 2/3 (30 mcg)";
               TRTDELTN=8;
            end;
      end;
run;
*******************************************************************************************;
* Output datasets *;
*******************************************************************************************;

Data datvout.adsl(label="Subject-Level Analysis Dataset");
   Retain StudyId UsubjId Subjid SiteId Age AgeU AAge AAgeU Sex SexN Race RaceN
      Ethnic EthnicN Country SAFFL RANDFL ENRLFL DTHFL Arm ArmCd ActArm ActArmCd
      TRT01P TRT01PN TRT02P TRT02PN TRT01A TRT01AN TRT02A TRT02AN TrtSeqP TrtSeqA
      BrthDt BrthDtF DthDtc DthDt DthDtF SrvLacDt EnrlDt EnrlNo RandDt RandNo
      RfStDt RfStTm RfEnDt RfEnTm RfpEnDt RfIcDt TrtSdt TrtStm TrtEdt TrtEtm
      TR01SDT TR01STM TR01EDT TR01ETM TR02SDT TR02STM TR02EDT TR02ETM TrtSdtm
      TrtEdtm TR01SDTM TR01EDTM TR02SDTM TR02EDTM vax101dt vax102dt vax201dt
      agetr01 agetru01 agetr02 agetru02 agetgr1n agetgr1 AAgeY AAgeYu AAgeM AAgeMu
      AAgeW AAgeWu AAgeD AAgeDu AAgeH AAgeHu AgeGr1 AgeGr1N Arace AraceN RaceGr1
      RaceGr1N Aethnic AethnicN EosDcDt EosDcRs;
   Set adsl;
```

Attrib TRT01P Label="Planned Treatment for Period 01" TRT01PN
Label="Planned Treatment for Period 01 (N)" TRT02P
Label="Planned Treatment for Period 02" TRT02PN
Label="Planned Treatment for Period 02 (N)" TRT01A
Label="Actual Treatment for Period 01" TRT01AN
Label="Actual Treatment for Period 01 (N)" TR01SDT
Label="Date of First Exposure in Period 01" TR01EDT
Label="Date of Last Exposure in Period 01" TR01STM
Label="Time of First Exposure in Period 01" TR01ETM
Label="Time of Last Exposure in Period 01" TR01SDTM
Label="Datetime of First Exposure in Period 01" TR01EDTM
Label="Datetime of Last Exposure in Period 01" TRT02A
Label="Actual Treatment for Period 02" TRT02AN
Label="Actual Treatment for Period 02 (N)" TR02SDT
Label="Date of First Exposure in Period 02" TR02EDT
Label="Date of Last Exposure in Period 02" TR02STM
Label="Time of First Exposure in Period 02" TR02ETM
Label="Time of Last Exposure in Period 02" TR02SDTM
Label="Datetime of First Exposure in Period 02" TR02EDTM
Label="Datetime of Last Exposure in Period 02" TrtSeqP
Label="Planned Sequence of Treatments" TrtSeqA
Label="Actual Sequence of Treatments" SafFl Label="Safety Population Flag"
RandFl Label="Randomized Population Flag" RandDt
Label="Date of Randomization" RandNo Label="Randomization Number" EnrlFL
Label="Enrolled Population Flag" EnrlDt Label="Date of Enrollment" EnrlNo
Label="Enrollment Number" RfStDt Label="Subject Reference Start Date" RfEnDt
Label="Subject Reference End Date" RfStTm
Label="Subject Reference Start Time" RfEnTm
Label="Subject Reference End Time" RfPEnDt
Label="Date of End of Participation" RfIcDt Label="Date of Informed Consent"
DthDt Label="Date of Death" DthDtf Label="Date of Death Imputation Flag"
TrtSdt Label="Date of First Exposure to Treatment" TrtEdt
Label="Date of Last Exposure to Treatment" TrtStm
Label="Time of First Exposure to Treatment" TrtEtm
Label="Time of Last Exposure to Treatment" TrtSdtm
Label="Datetime of First Exposure to Treatment" TrtEdtm
Label="Datetime of Last Exposure to Treatment" AgeGr1
Label="Pooled Age Group 1" AgeGr1N Label="Pooled Age Group 1 (N)" AgeGr4
Label="Pooled Age Group 4" AgeGr4N Label="Pooled Age Group 4 (N)" BrthDtf
Label="Date of Birth Imput. Flag" BrthDt Label="Date of Birth" AAge
Label="Analysis Age" AAgeU Label="Analysis Age Unit" AgeGr2
Label="Pooled Age Group 2" AgeGr2N Label="Pooled Age Group 2 (N)" AgeGr3
Label="Pooled Age Group 3" AgeGr3N Label="Pooled Age Group 3 (N)" AAgeY
Label="Analysis Age in Years" AAgeM Label="Analysis Age in Months" AAgeW
Label="Analysis Age in Weeks" AAgeD Label="Analysis Age in Days" AAgeH
Label="Analysis Age in Hours" AAgeYu Label="Analysis Age in Years Units"
AAgeMu Label="Analysis Age in Months Units" AAgeWu
Label="Analysis Age in Weeks Units" AAgeDu Label="Analysis Age in Days Units"
AAgeHu Label="Analysis Age in Hours Units" SexN Label="Sex (N)" RaceN
Label="Race (N)" EthnicN Label="Ethnicity (N)" Arace Label="Analysis Race"
AraceN Label="Analysis Race (N)" Aethnic Label="Analysis Ethnicity" AethnicN
Label="Analysis Ethnicity (N)" RaceGr1 Label="Pooled Race Group 1" RaceGr1N
Label="Pooled Race Group 1 (N)" EosDcDt
Label="End Of Study Discontinuation Date" Format=date9. EosDcRs

```
    Label="End Of Study Discontinuation Reason";
  Drop _: Derived_: AAGE: AETHNIC: AGETGR1: AGETR02 AGETRU02 BRTHDTC EVSAD2FL
    EXCRIT5 BE1DT: BE2DT: BE3DT: BE4DT: CODT: COVIS: DSRANGRP EFFICACY ENRLDT
    ENRLNO IMMUNO INEX ISDT: MULTIPLE RACE1-RACE2 SAFETY SITEEXCLD w3pv1dt
    SRT1-SRT5 SRVLACDT TMPDTC TMPID VAX101DTM VAX102DTM VAX103DTM VAX201DTM
    VAX202DTM VAX203DTM VIS: STEXCFL UNKRDFL EV1MEFFL m1pv3dt m1pv4dt
    BE3_1DT: BE4_1DT: V3DATE V3P2 V5P1 PERIOD1_POP3 Domain RfxStDtc RfxEnDtc
    RfxStDt RfxEnDt RfxStTm RfxEnTm RfStDtc RfEnDtc RfIcDtc RfPEnDtc cdecase;
Run;

proc printto;
run;
```