

```

*****
*****;

** Program Name      : adsl.sas
                      **;

** Date Created      : 07Mar2021
                      **;

** Programmer Name:  (b) (6)
                      **;

** Purpose           : Create adsl dataset
                      **;

** Input data        : dm suppdm ex suppex ds suppds is co lb cm ie dv suppdv vs sv mb
suppmb mh pr          **;

**
                      face ce ho suppho
                      **;

** External file     : ../prjC459/nda2_unblinded_esub/euaext_esub_adam/saseng/cdisc3_0/data
                      **;

** Output data       : adsl.sas7bdat
                      **;

*****
*****;

%let
oprot=/Volumes/app/cdars/prod/sites/cdars4/prjC459/nda2_unblinded_esub/bla_euaext_esub_sdt
m/saseng/cdisc3_0;

%let
prot=/Volumes/app/cdars/prod/sites/cdars4/prjC459/nda2_unblinded_esub/euaext_esub_adam/sas
eng/cdisc3_0;

libname dataprot "&oprot./data" access=readonly;

libname datvprot "&prot./data_vai";

%*Path for external files;

%let expath=&prot./data;

%*Insert the date of snapshot;

%let cutoff2=13Mar2021;

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

*****

Clean *;

*****

```

090477e196b7a9f9\Final On: 06-Apr-2021 21:45 (GMT)

```

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 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;

```

090177e196b7a9f0\Final\Final On: 06-Apr-2021 21:45 (GMT)

```

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")) 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

```

090177e196b7a9f9\Final\Final On: 06-Apr-2021 21:45 (GMT)

```

when (calculated period)="Double Blinded Period" then 1 when (calculated
period)="Open Label Period" then 2 end as PERIODN, case when not
missing(a.extrtrt) then a.extrtrt else "ZZZZZZ" end as _extrtrt_ from ExSet a left
join prd2 b on a.usubjid=b.usubjid order by periodn, period, usubjid, _extrtrt_;

quit;

data ExSet;

set ExSet;

by periodn period usubjid _extrtrt_;

retain ACTDRUG;

if first.usubjid then

    ACTDRUG=extrtrt;

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.tmpdttc, d.qval as

DSRANGRP from DmSet a left join randcode b on a.usubjid=b.usubjid left

join (select distinct usubjid, max(exstdttc) as tmpdttc from ExSet where not

missing(exstdttc)) 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;

```

00177e196b7a9f0\Final\Final On: 06-Apr-2021 21:45 (GMT)

```
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;
```

090177e196b7aef0\Final\Final On: 06-Apr-2021 21:45 (GMT)

```
        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;
```

090477e496b7a9f0Final\Final On: 06-Apr-2021 21:45 (GMT)

```
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_suppsds1 (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_suppsds1 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.supplex;

    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_suppsds1 (drop=idvar idvarval _NAME_ _LABEL_);

    set _spmdel_supp_dsin_idvar1_h;

    if idvar="EXSEQ";

```

090177e196b7a9f0\Final On: 06-Apr-2021 21:45 (GMT)



```

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";

```

090177e196b7a9f0\Final On 06-Apr-2021 21:45 (GMT)

```

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_suppsds1 (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_suppsds1 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;

```

090177e196b7a9f0\Final On: 06-Apr-2021 21:45 (GMT)

```

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;

```

090177e196b7a9f0\Final\Final On: 06-Apr-2024 21:45 (GMT)

```
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;
            end;
    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=.;
```

090177e196b7a9f0\Final\Final On# 06-Apr-2021 21:45 (GMT)

```
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 \*;

090477e196b7a9f0\Final\Final On: 06-Apr-2021 21:45 (GMT)

```

*****
;

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;

```

090177e196b7a9fa\Final On 06-Apr-2021 21:45 (GMT)



```

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="TrtNFFmt";

    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

```

090177e196b7a9f0\Final\Final On 06-Apr-2021 21:45 (GMT)

090177e196b7a9f0\Final\Final On: 06-Apr-2021 21:45 (GMT)

```
        _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;

090177e196b7a9f9\Final On 06-Apr-2021 21:45 (GMT)

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));

090177e196b7a9f0\Final\On: 06-Apr-2021 21:45 (GMT)

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=Comp1Dt))

NoDupKey;

By UsubjId;

Where Upcase(Strip(DsCat))="DISPOSITION EVENT" and

090177e196b7a9fa9f06-Apr-2021 21:45 (GMT) Final On: 06-Apr-2021 21:45 (GMT)

090177e496b7a9f0\Final\Final On: 06-Apr-2021 21:45 (GMT)

```
        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;

```

090477e496b7a9f0\Final\Final On: 06-Apr-2021 21:45 (GMT)



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 datasets. *;  
* 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";

090177e196b7a9f0\Final\Final On 06-Apr-2024 21:45 (GMT)

```
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 ("B1_10", "B1_100", "B1_20", "B1_30", "B2_10",
            "B2_20", "B2_30", "B2_P23_30", "PLACEBO") then
            do;
                ActArmcd=derived_actarmcd;
                Actarm=derived_actarm2;
            end;
        end;
    end;

```

```

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 not in ("B1_10", "B1_100", "B1_20", "B1_30",
    "B2_10", "B2_20", "B2_30", "B2_P23_30", "PLACEBO") and not
    missing(derived_actarmcd) then
    do;
        ActArmCd="NOTTRT";
        ActArm="Not Treated";
    End;

Else if derived_actarmcd not in ("B1_10", "B1_100", "B1_20", "B1_30",
    "B2_10", "B2_20", "B2_30", "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 Race0th 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;

```

090177e196b7a9f0\Final\Final On: 06-Apr-2021 21:45 (GMT) :-

```
_birthday=day(brthdt);
_birhtmth=month(brthdt);
_birthyr=year(brthdt);
_Enrldtday=day(Enrldt);
_Enrldtmth=month(Enrldt);
_Enrldtyr=year(Enrldt);
aage=_Enrldtyr - _birthyr;

if (_Enrldtmth lt _birhtmth) or ((_Enrldtmth eq _birhtmth) 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);
_birthdaymth=month(brthdt);
_birthdayyr=year(brthdt);
_Enrldtday=day(Enrldt);
_Enrldtmth=month(Enrldt);
_Enrldtyr=year(Enrldt);
aagey=_Enrldtyr - _birthdayyr;

if (_Enrldtmth lt _birthdaymth) or ((_Enrldtmth eq _birthdaymth) 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;

```

090177e196b7a9f0\Final\Final On: 06-Apr-2021 21:45 (GMT)



0900177e196b7a9f0\Final\Final On: 06-Apr-2024 21:45 (GMT)

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

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

if index(vst_org, "_VAX4") 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");

```

090177e196b7a9f0\Final\Final On: 06-Apr-2021 21:45 (GMT)

```
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);
```

090177e196b7a9f0\Final\Final On: 06-Apr-2021 21:45 (GMT)

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

    if n(agetru02) then
        agetru02="YEARS";
```

```
format vax102dt date9.;
attrib agetru03 label="Age Units at Vaccination 03" length=$6;

if n(agetru03) then
    agetru03="YEARS";
format vax103dt date9.;
format vax201dt date9.;
format vax202dt date9.;
format vax203dt date9.;
length INFAGE agetgr1 $40.;

If 12<=(agetru01)<=15 and missing(dsrangrp) then
    INFAGE='12-15 Years';
Else If 16<=(agetru01)<=55 and missing(dsrangrp) then
    INFAGE='16-55 Years';
Else If 18<=(agetru01)<=55 and not missing(dsrangrp) then
    INFAGE='18-55 Years';
Else If 65<=(agetru01) and not missing(dsrangrp) then
    INFAGE='65-85 Years';
Else If 56<=(agetru01) and missing(dsrangrp) then
    INFAGE='>55 Years';
agetgr1=INFAGE;
agetgrln=input(INFAGE, ??INFAGE.);
attrib agetgr1 label="Age Group at Vaccination 01" agetgrln
    label="Age Group at Vaccination 01 (N)";
run;

data adsl;
merge adsl(in=a) exvac2(keep=usubjid vax101dt vax102dt vax103dt vax201dt
    vax202dt vax203dt agetru01 agetru02 agetru03 agetru03 agetru03 agetgrln
    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;

```

090177e196b7a9f0\Final\Final On: 06-Apr-2021 21:45 (GMT)

```

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

data srv_lb;
    set srv_lb;
    by usubjid lbd;

    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;

```

090177e196b7a9fa\Final On: 06-Apr-2021 21:45 (GMT)

```

*****
;

* PR *;

*****
;

data srv_pr;

    set dataprot.pr;

    prstdt=input(prstdtc, ??is860lda.);

    format prstdt date9.;

    prendt=input(prendtc, ??is860lda.);

    format prendt date9.;

    _maxprdt=max(prstdt, prendt);

    keep usubjid _maxprdt;

run;

proc sort data=srv_pr;

    by usubjid _maxprdt;

run;

data srv_pr;

    set srv_pr;

    by usubjid _maxprdt;

    if last.usubjid;

run;

*****

DS *;

*****

data srv_ds;

    set dataprot.ds;

    dsstdt=input(dsstdtc, ??is860lda.);

    format dsstdt date9.;

    where dsdecod not in ("LOST TO FOLLOW-UP", "DEATH", "ENROLLED");

```

090177e196b7a9f0\Final\On: 06-Apr-2021 21:45 (GMT)



```

        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';

```

090177e196b7a9f0\Final\Final On: 06-Apr-2021 21:45 (GMT+8)

090177e196b7a9f0\Final\Final On: 06-Apr-2024 21:45 (GMT)

```
        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_pr srv_ds;
    by usubjid;

    if a;
```

0901727e196b7a9f0\Final\Final On: 06-Apr-2024 21:45 (GMT)

```
_srvlacdt=max(trtsdt, trtedt, RANDDT, vsdt, lbdtd, _maxcmdt, _maxprdt, 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. *;
```

```
*****
```

```
Cohort info;
```

```
data suppds;
```

```
set dataprot.suppds;
```

```
where qnam="DSRANGRP";

proc sort;
  by usubjid;
run;

data adsl;
  merge adsl(in=a) suppbs(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;
```

090177e196b7a9f9\Final On 06-Apr-2021 21:45 (GMT)

```
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 1 & 2 & 3;

*proc import datafile="&expath./phase1-participants.xlsx" out=phase1 dbms=xlsx
replace;
/*EXLX;
setnames=yes;
run;*/

/*Check file name before finalization;
```

```

proc import datafile="&expath./C4591001_Phase 1 subjects from DMW.xlsx"
    out=phase1 dbms=xlsx replace;
    getnames=yes;
run;

proc sql UNDO_POLICY=NONE;
    create table phase1 as select strip(put(SUBJECTNUMBERSTR, best.)) as subjid
        from phase1 order by subjid;
quit;

/*proc import datafile="&expath./phase2-360-participants.xlsx" out=phase2
dbms=xlsx replace;
RXLX;
getnames=no;
datarow=2;
sheet="SubjID_360";
run;*/

*Check file name before finalization;
proc import
datafile="&expath./first_C4591001_360_participants_enrolled_V1.0_13Aug2020_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./phase3-6k-participants.xlsx" out=phase3

```

090177e1966b7a9f0\Final\Final On: 06-Apr-2021 21:45 (GMT)

```

dbms=xlsx replace;

RXLX;

getnames=no;

datarow=2;

run;*/

*Check file name before finalization;

proc import
datafile="%expath./newlist_C4591001_6k_participants_enrolled_V3.0_17sep2020.csv"

    out=phase3 dbms=csv replace;

    getnames=no;

    datarow=2;

run;

data phase3;

    set phase3;

    length subjid $20;

    subjid=scan(var1, -1, " ");

    *Check file name before finalization;

    *subjid=scan(A, -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";

```

090177e196b7a9f0\Final\Final On: 06-Apr-2021 21:45 (GMT)

```
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;
end;

if phasen ne 1 then
  do;
    dosplvl="";
    dosalvl="";
    dosplvln=.;
    dosalvln=.;
  end;
run;

Categorize age groups;

data adsl;
  set adsl;

  if missing(AGETR01) then
    do;
      AGETR01=AAGE;
      AGETRU01=AAGEU;
```



```
    AGETR1=AGEGR1;
    AGETR1N=AGEGR1N;
end;
AGEGR1=AGETR1;
AGEGR1N=AGETR1N;
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 16<=agetr03<=55 then
            do;
                agetgr3="16-55 Years";
                agetgr3N=1;
            end;
        else if agetr03>55 then
            do;
                agetgr3=">55 Years";
                agetgr3N=2;
            end;
        end;
    else
        do;
            agetr03=.;
            agetru03="";
        end;
    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;

    extptrefln=periodn*100+extmp;

    extptrefl="VAX"||strip(put(extptrefln, best.));

    format extime time8. exdatetime datetime20.;

    extime=input(scan(exstdtc, 2, "T"), time8.);

    exdatetime=input(exstdtc, is8601dt.);

```

090177e196b7a9f0\Final\Final On: 06-Apr-2021 21:45 (GMT)

090177e196b7a9f0\Final\Final On: 06-Apr-2021 21:45 (GMT)

```
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)";

if extrt="Placebo" then
    extrt1=extrt;

proc sort;
    by usubjid extptrefln;
run;

proc transpose data=ex out=t_ex;
    by usubjid;
    id extptrefl;
    var extrt1;
run;

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

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

\*Correct Period 2 info;

proc format;

```
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;
```

run;

```
data ads1(rename=(VAX103=VAX10U VAX203=VAX20U VAX103DT=VAX10UDT
                  VAX203DT=VAX20UDT VAX103TM=VAX10UTM VAX203TM=VAX20UTM));
merge ads1(in=a drop=trtsepp trtsega 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 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);
```

090177e196b7a9f0\Final\Final On: 06-Apr-2021 21:45 (GMT)

```

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;
TRT01P=_TRT01P;
TRT01PN=input(TRT01P, trtfmt.);
end;

if _TRT01A ne "" then
do;
if _TRT01A="Placebo" then
TRT01A=_TRT01A;
else if PHASEN=1 and index(_TRT01A, "BNT") and index(dosalv1, "/")=0 then
TRT01A=strip(_TRT01A)||" Phase 1 ("||tranwrd(scan(strip(dosalv1), 1,
"/"),
"(*ESC*){unicode 03BC}g", "mcg")||")";
else if PHASEN=1 and index(_TRT01A, "BNT") then
TRT01A=strip(_TRT01A)||" Phase 1 ("||strip(scan(scan(dosalv1, 1, "/"), 1,
"(*ESC*)"))||"/"||strip(scan(scan(dosalv1, 2, "/"), 1,
"(*ESC*)"))||" mcg)";
else if index(_TRT01A, "BNT") then
TRT01A=strip(_TRT01A)||" Phase 2/3 (30 mcg)";
TRT01AN=input(TRT01A, trtfmt.);
end;

```

```

if tr02sdt>. then
    do;

        if PHASEN=1 then
            TRT02P="BNT162b2 Phase 1 (30 mcg)";
        else
            TRT02P="BNT162b2 Phase 2/3 (30 mcg)";
        TRT02PN=input(TRT02P, trtfmt.);

        if PHASEN=1 and (index(VAX201, "BNT") or index(VAX202, "BNT") or
            index(VAX203, "BNT")) then
            TRT02A="BNT162b2 Phase 1 (30 mcg)";
        else if (index(VAX201, "BNT") or index(VAX202, "BNT") or index(VAX203,
            "BNT")) then
            TRT02A="BNT162b2 Phase 2/3 (30 mcg)";
        TRT02AN=input(TRT02A, trtfmt.);
    end;

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

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

*****
Specification 3 *;
ADD PERIOD 2 DISPOSITION INFO *;
1 - UNBLNDDT/REVMCDT. *;
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

```



000177e196b7a9f0\Final\Final On 06-Apr-2021 21:45 (GMT)

```
UNBLNDDT format=date9. label="Treatment Unblinded Date" from DsSet where
DSDECOD="TREATMENT UNBLINDED" and DSSTDTC ne "" order by usubjid;

quit;

data adsl;
    merge adsl _unblnd;
    by usubjid;
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 dsicd(keep=usubjid dsstdt rename=(dsstdt=REVMICDT));

label REVMICDT="Re-vax Informed Consent Date";

by usubjid;

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 - Comorbidities flag. *;

3 - COVID baseline info. *;

```

090477e196b7a9f0\Final\Final On: 06-Apr-2021 21:45 (GMT)

```
*****  
;  
*Add baseline BMI category;  
  
data bmi;  
    set dataprot.vs;  
    where index(visit, "V1_DAY1") 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;
```

090177e196b7a9f0Final\Final On: 06-Apr-2021 21:45 (GMT)

```
else if BMICAT="Missing" then
    BMICATN=5;

run;

*Check file name before finalization;

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<obscur then
    OBESEFL="N";
else if vsstresn>=obscur then
    OBESEFL="Y";

run;

*ADSL for Comorbidities - Xstart;

/*
proc import datafile="&expath./report-cci-periph-vasc.xlsx" out=fileout
    dbms=xlsx replace;
run;
data out;
set fileout(drop=a);
run;

proc import datafile="&expath./report-cci-hemiplegia.xlsx" out=fileout
    dbms=xlsx replace;
run;
data out;

```

090177e1-196b-7a9f0-06-Apr-2021 21:45 (GMT)

```

set out fileout(drop=a);
run;

proc import datafile="&expath./report-cci-lymphoma.xlsx" out=fileout dbms=xlsx
replace;
RXLX;
datarow=17;
getnames=yes;
run;

data out;
set out fileout(drop=a);
run;

proc import datafile="&expath./report-cci-leukemia.xlsx" out=fileout dbms=xlsx
replace;
RXLX;
datarow=17;
getnames=yes;
run;

data out;
set out fileout(drop=a);
run;

proc import datafile="&expath./report-cci-mod-sev-liver.xlsx" out=fileout
dbms=xlsx replace;
RXLX;
datarow=17;
getnames=yes;
run;

data out;
set out fileout(drop=a);
run;

proc import datafile="&expath./report-cci-aids-hiv.xlsx" out=fileout dbms=xlsx

```

090177e196b7a9f0FinalFinal  
06-Apr-2024 21:45 (GMT)







```

set out fileout(drop=a);

run;

proc import datafile="%expath./report-cci-mild-liver.xlsx" out=fileout
dbms=xlsx replace;

RXLX;

datarow=17;

getnames=yes;

run;

data out;

set out fileout(drop=a);

run;

proc import datafile="%expath./report-cci-diabetes-without-comp.xlsx"
out=fileout dbms=xlsx replace;

RXLX;

datarow=17;

getnames=yes;

run;

data out;

set out fileout(drop=a);

run;

proc import datafile="%expath./report-cci-cerebrovascular.xlsx" out=fileout
dbms=xlsx replace;

RXLX;

datarow=17;

getnames=yes;

run;

data out;

set out fileout(drop=a);

run;

proc import datafile="%expath./report-cci-mi.xlsx" out=fileout dbms=xlsx

```



```
getnames=yes;

run;

data out;

set out fileout(drop=a);

run;

*/

*Check file name before finalization;

%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(name, "Report_CCI") 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;

    %end;

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;

%*End;
%*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;

data is_rep;
    set is;

    if not (phasen=1 and cohortn in (1.18 1.38) and ISTSTDTL^="REPEAT TEST" and
        visitnum in (60748 60751 60754 60755));
run;

```

090177e196b7a9fa\Final\Final On: 06-Apr-2021 21:45 (GMT)

```

data is;
    set is_rep;
    where index(visit, "V1_DAY1_") and istested 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_") and mbtested='RTCOV2NS';

```

090177e1-196b7a-9f0\Final\Final On: 06-Apr-2021 21:45 (GMT)

```

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;

```

090177e196b7a9f0\Final\Final On: 06-Apr-2021 21:45 (GMT)

090177e196b7a9f0\Final\Final On 06-Apr-2021 21:45 (GMT)

```
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 cocovall;
    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";
run;

proc sql;
    create table cocovall2 as select a.*, b.vax102dt, b.vax10Udt from cocovall a
        left join adsl b on a.usubjid=b.usubjid;
quit;

data cocovall3;
    set cocovall2;

    if not missing(codtc) and not missing(vax102dt) then
```



```

do;
    d2diff=input(codtc, yymmdd10.)-vax102dt;

    if 28<=d2diff<=42 then
        chflg=1;
    chabs=abs(d2diff-30);

    if 42<d2diff then
        ch2flg=1;
    ch2abs=abs(d2diff-190);
end;

run;

data is;
    set is_rep;
    where (index(visit, "COVID") or upcase(strip(visit))='V101_VAX3' or
        upcase(strip(visit))='V201_SURVEIL_CONSENT') and isorres ne "";

proc sort nodupkey;
    by usubjid visitnum;
run;

proc sql;
    create table cocovall4 as select a.*, b.isorres from
        cocovall3(where=(chflg=1)) a left join is b on a.usubjid=b.usubjid and
        a.visitnum=b.visitnum order by usubjid, chabs, isorres desc;
quit;

proc sort data=cocovall4 nodupkey;
    by usubjid;
run;

proc sql;
    create table cocovall5 as select a.*, b.isorres from
        cocovall3(where=(ch2flg=1)) a left join is b on a.usubjid=b.usubjid and
        a.visitnum=b.visitnum order by usubjid, ch2abs, isorres desc;
quit;

```

090177e196b7a9f0\Final\Final On 06-Apr-2021 21:45 (GMT)

```

data cocovchk;
    set cocovall5;
run;

proc sort data=cocovall5 nodupkey;
    by usubjid;
run;

proc sql;
    create table cotemp as select a.*, b.visit as covvis, b.visitnum as covvisnum,
        b.codtc as covdte, b.cody as covdy from dataprot.co a left join cocovall4 b
        on a.usubjid=b.usubjid;
quit;

proc sql;
    create table cotemp2 as select a.*, b.visit as covvis, b.visitnum as
        covvisnum, b.codtc as covdte, b.cody as covdy from dataprot.co a left join
        cocovall5 b on a.usubjid=b.usubjid;
quit;

data subcov;
    set cotemp(where=(not missing(covvis) and index(visit, "_MONTH1_") and
        COREF='Sample Collected' and COVAL="N")) cotemp2(where=(not missing(covvis)
        and index(visit, "_MONTH6_") and COREF='Sample Collected' and COVAL="N"));
    mvis=visit;
    mvisnum=visitnum;
    mdte=codtc;
    keep usubjid mvis mvisnum mdte covvis covvisnum covdte;
run;

proc sql;
    create table conew0 as select a.*, b.mvis, b.mvisnum, b.mdte, b.covvis,
        b.covvisnum, b.covdte from dataprot.co a left join subcov b on
        a.usubjid=b.usubjid and a.visitnum=b.covvisnum and a.codtc=b.covdte;
quit;

```

090177e196b7a9f0\Final\Final On: 06-Apr-2021 21:45 (GMT)

```

data conew;
  set conew0;

  if not missing(covvisnum) then
    do;
      visit=mvis;
      visitnum=mvisnum;
      codtcsave=mdtc;
    end;
  else
    codtcsave=codtc;

run;

proc sql;
  create table isnew0 as select a.*, b.mvis, b.mvisnum, b.mdtc, b.covvis,
    b.covvisnum, b.covdtc from is_rep a left join subcov 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) then
    do;
      visit=mvis;
      visitnum=mvisnum;
    end;

run;

proc sql undo_policy=none;
  create table newis as select coalesce(a.usubjid, b.usubjid) as usubjid,
    coalesce(a.visitnum, b.visitnum) as visitnum, coalesce(a.visit, b.visit) as
    visit, a.DOMAIN, a.EPOCH, a.ISBLFL, a.ISCAT, a.ISDTC, a.ISDY, a.ISGRPID,
    a.ISLLOQ, a.ISMETHOD, a.ISORRES, a.ISORRESU, a.ISREFID, a.ISSEQ, a.ISSPEC,
    a.ISSTRESC, a.ISSTRESN, a.ISSTRESU, a.ISTEST, a.ISTESTCD, a.STUDYID,

```

090177e196b7a9f0\\Final\\Final On: 06-Apr-2021 21:45 (GMT+8)

090177e496b7a9f0\Final\Final On: 06-Apr-2021 21:45 (GMT)

```
a.covdttc, a.covvis, a.covvisnum, a.mdtc, a.mvis, a.mvisnum, b.COVAL, b.CODTC,
b.codtcsave from isnew a full join conew(where=(RDOMAIN="IS" and
COREF="Sample Collected" and COVAL="Y")) b on a.usubjid=b.usubjid and
a.visitnum=b.visitnum;

create table newis as select a.*, b.cohortn, b.vax101dt, b.vax102dt,
b.vax10Udt from newis a left join adsl b on a.usubjid=b.usubjid order by
usubjid, visitnum, isdtc, codtc, codtcsave;

quit;

data isva;

set newis;

format isdate codate bedt date9.;

isdate=input(isdtc, yymmdd10.);
codate=input(codtc, yymmdd10.);
bedt=input(codtcsave, yymmdd10.);

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;
cdiff=codate-vax102dt;
end;
else if not missing(vax101dt) then
do;
novax=1;
diff=isdate-vax101dt;
cdiff=codate-vax101dt;
end;
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 sort data=isva out=v0xdt nodupkey;

    by usubjid visitnum isdttc codttc codttcsave;

    where not (cohortn=1.16 and visitnum in (60751 60752 60753 60754));

run;

data adsl;

    merge adsl (in=a) v0xdt(keep=usubjid bedt visit cohortn rename=(bedt=bedt2)
        where=(index(visit, "V103_")=0 and ((index(visit, "_MONTH1_") and cohortn ne
        1.16) or
            (visit="V7_MONTH1_S_R") and cohortn=1.16)))
        v0xdt(keep=usubjid bedt visit cohortn rename=(bedt=bedt2) where=(index(visit,
        "V104_")=0 and index(visit, "_MONTH6_"))) v0xdt(keep=usubjid bedt visit
        cohortn rename=(bedt=bedt2) where=(visit in ("V103_MONTH1")))
        v0xdt(keep=usubjid bedt visit cohortn rename=(bedt=bedt2) where=(visit
        in ("V104_MONTH6"))) sv(keep=usubjid svstdt visit cohortn
        rename=(svstdt=bedt2) where=(index(visit, "V103_")=0 and ((index(visit,
        "_MONTH1_") and cohortn ne 1.16) or
            (visit="V7_MONTH1_S_R") and cohortn=1.16)))
        sv(keep=usubjid svstdt visit cohortn rename=(svstdt=bedt2)
        where=(index(visit, "V104_")=0 and index(visit, "_MONTH6_"))) sv(keep=usubjid
        svstdt visit cohortn rename=(svstdt=bedt2) where=(visit in ("V103_MONTH1")))
        sv(keep=usubjid svstdt visit cohortn rename=(svstdt=bedt2) where=(visit
        in ("V104_MONTH6")));

```

090177e196b7a9f0\Final\Final On: 06-Apr-2021 21:45 (GMT)

```
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 Vax4" format=date9.
    V04DT label="Date of Visit at 6M after Vax4" 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 arm="Placebo" and (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 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;

un;

\*\*\*\*\*

```

;
* 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=col(keep=usubjid visitnum visit coval) nodupkey;
    by usubjid visitnum;
run;

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

proc sort data=isva out=co2(keep=usubjid codate visitnum visit) nodupkey;
    by usubjid visitnum;
run;

proc transpose data=co2 out=t_co2 prefix=codt;
    by usubjid;
    id visitnum;
    idlabel visit;
    var codate;
run;

proc sort data=isva out=co3(keep=usubjid cdiff visitnum visit) nodupkey;
    by usubjid visitnum;
run;

proc transpose data=co3 out=t_co3 prefix=cdiff;

```

090177e196b7a9f9\Final\Final On: 06-Apr-2021 21:45 (GMT)



```

by usubjid;
id visitnum;
idlabel visit;
var cdiff;

run;

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

run;

proc transpose data=isval out=t_isval 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=isd;
by usubjid;
id visitnum;
idlabel visit;
var isdate;

run;

proc sql;
create table dv as select a.usubjid, a.dvstdtc, a.dvseq, 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;
create table dvdate as select distinct usubjid, min(input(dvstdtc, yymmdd10.))

```

090177e196b7a9f0\Final On: 06-Apr-2021 21:45 (GMT)

090177e1966b7a9f9\Final\Final On: 06-Apr-2021 21:45 (GMT)

```
as dvstdt label="Start Date of Important PD" format=date9.
from dv where dvstdtc ne "" group by usubjid;
create table dvout 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 siteexcl 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;
quit;

proc sort data=dataprot.ie out=ie(keep=usubjid domain) nodupkey;
by usubjid;
where visit ne "V101_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 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 incl8p incl3p;
    set isva;

    if istestcd^="C19NIG" and isorres not in (" " "IND" "QNS" "NOT DONE")
        and (visit="V8_MONTH6_S" or (visit in ("V5_WEEK1_POSTVAX2_S_R"
        "V6_WEEK2_POSTVAX2_S_R" "V7_MONTH1_S_R") and cohortn=1.16) or
        (visit in ("V3_MONTH1_POSTVAX2_L" "V4_MONTH6_L" "V5_WEEK1_POSTVAX2_S"
        "V6_MONTH24_L" "V6_WEEK2_POSTVAX2_S" "V7_MONTH1_S") and cohortn ne 1.16)) then
        output incl8p;
    else if istestcd^="C19NIG" and isorres not in (" " "IND" "QNS" "NOT DONE")
        and (visit="V3_WEEK1_POSTVAX1_S" or (visit="V4_WEEK3_VAX2_S_R" and
        cohortn=1.16) or (visit="V4_WEEK3_VAX2_S" and cohortn ne 1.16)) then
        output incl3p;

run;

proc sort data=incl3p nodupkey;
    by usubjid;
run;

proc sort data=incl8p nodupkey;
    by usubjid;
run;

data adsl;
merge adsl(in=a) ie(rename=(domain=INEX)) t_isva1 t_isva2 t_co1 t_co2 t_co3
    nbind sv(keep=usubjid visit svstdt cohortn rename=(svstdt=visit3dt)
    where=(index(visit, "V103_")=0 and ((index(visit, "_MONTH1_") and cohortn ne
    1.16) or
        (visit="V7_MONTH1_S_R") and cohortn=1.16)))
    dvout (keep=usubjid dvstdt safety efficacy immuno multiple siteexcl)
    incl3p(in=incl3p keep=usubjid) incl8p(in=incl8p keep=usubjid);
by usubjid;

if a;

attrib BLDV1FL label="Blood Sample Drawn before Vax 1" BLDV2FL

```

090177e196b7a9f0\Final\Final On: 06-Apr-2024 21:45 (GMT)

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 BD within the timeframe 2" INCL10FL

label="Unblinded after 1M post Dose 2 visit" 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. 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" AAIEFFL  
 label="Dose 1 all-available Efficacy Popu Flag" AAI2EFFL  
 label="Dose 2 all-available Efficacy Popu Flag";

```

if not missing(siteexcl) then
  RSEXSAF="Unreliable data due to lack of PI oversight";
else if not missing(safety) then
  RSEXSAF="Did not provide informed consent";
if not missing(covis60748) then
  BLDV1FL="Y";
else 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
    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
    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(covis60754) and cohortn^=1.16 then
    BLDV6FL="Y";
else if not missing(covis1165457) and cohortn=1.16 then
    BLDV6FL="Y";
else if not missing(covis60767) then
    BLDV6FL="Y";
else
    BLDV6FL="N";

if not missing(covis60755) or not missing(covis1165458) or not
missing(covis60768) then
    BLDV7FL="Y";
else
    BLDV7FL="N";

if not missing(codt60748) then
    BLDV1DT=codt60748;
else 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;

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;

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;

if not missing(codt1165457) and cohortn=1.16 then
    BLDV6DT=codt1165457;

if not missing(codt60767) then
```

```
BLDV6DT=codt60767;

if not missing(codt60755) then
    BLDV7DT=codt60755;

if not missing(codt1165458) and cohortn=1.16 then
    BLDV7DT=codt1165458;

if not missing(codt60768) then
    BLDV7DT=codt60768;

if not missing(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(vis60751) and VAX101DT>. and phasen=1 then
    INCL4FL="Y";
else if phasen=1 then
    INCL4FL="N";

if 19<=cdiff60751<=23 and phasen=1 then
    INCL5FL="Y";
else if phasen=1 then
    INCL5FL="N";

if not missing(BLDV6DT) then
    visit3dt=BLDV6DT;

if not missing(safety) or (not missing(efficacy) and ((dvstdt-vax102dt<14)
    or (dvstdt-vax102dt>=14 and (dvstdt<=visit3dt)))) or not missing(immuno) then
    INCL6FL="N";
else
    INCL6FL="Y";

if 19<=VAX102DT-VAX101DT<=42 and vax101=vax102 and ARM ne "" /*and vax10udt=.*//
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 phasen=1 and VAX102DT>. and ((6<=cdiff60752<=8 and (codt60752<=vax201dt or  
vax201dt=.) and cohortn ne 1.16) or
```

```
(6<=cdiff1165455<=8
```

```
and (codt1165455<=vax201dt or vax201dt=.) and cohortn=1.16)) then
```

```
INCL9FL="Y";
```

```
else if phasen ne 1 and VAX102DT>. and (28<=cdiff60767<=42
```

```
and (codt60767<=vax201dt or vax201dt=.) then
```

```
INCL9FL="Y";
```

```
else
```

```
INCL9FL="N";
```

```
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 INCL4FL="N" then
                    do;
                        EXCL4FL="Y";
                        EXCRIT4="did not have at least 1 valid and determinate
immunogenicity result 21 days after Dose 1";
                    end;

                if INCL5FL="N" then
                    do;
                        EXCL5FL="Y";
                        EXCRIT5="did not have blood collection within 19-23 days
after Dose 1";
                    end;
                end;
            end;
        end;
    end;
end;
```

```

        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;

        if INCL9FL="N" and not missing(VAX102DT) then
            do;
                EXCL9FL="Y";

                if phasen^=1 then
                    EXCRIT9="did not have blood collection within 28-42 days
after Dose 2";

                else if phasen=1 then
                    EXCRIT9="did not have blood collection within 6-8 days
after Dose 2";

            end;
        end;
    end;

if INCL6FL="N" then
    do;
        length _ttt_ $100;

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

        if not missing(efficacy) then
            do;

                if not missing(vax102dt) then
                    do;

                        if .<dvstdt-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<=dvstdt-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 dvstdt-vax102dt>=14 and (.<dvstdt<=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 not missing(_ttt_) then

                _ttt_=strip(_ttt_)||", Immunogenicity";

            else

                _ttt_="Immunogenicity";

            end;

        EXCL6FL="Y";

        EXCRIT6="had important protocol deviation(s) as determined by the clinician

```

```
for "||strip(_ttt_)||" Population(s)";
    end;

if phasen=1 and INCL1fl="Y" and INCL2fl="Y" and INCL4fl="Y" and INCL5fl="Y"
    and saffl="Y" and missing(immuno) then
    EVAL01FL="Y";
else if phasen=1 then
    EVAL01FL="N";

if INCL1fl="Y" and INCL2fl="Y" and (INCL7fl="Y" and vax10udt=.) and
    INCL8fl="Y" and saffl="Y" and INCL9fl="Y" and missing(immuno) then
    EVAL02FL="Y";
else
    EVAL02FL="N";

if phasen=1 and randfl="Y" and saffl="Y" and (VAX101DT>. or vax102dt>.) and
    INCL3fl="Y" and not (not missing(immuno) and not missing(siteexcl)) then
    AAI01FL="Y";
else if phasen=1 then
    AAI01FL="N";

if randfl="Y" and saffl="Y" and vax102dt>. and INCL8fl="Y" and not (not
    missing(immuno) and not missing(siteexcl)) 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 dvstdt-vax102dt<7) and
    not (not missing(efficacy) and not missing(siteexcl)) then
    EVALEFFL="Y";
else
    EVALEFFL="N";
```

```

if randfl="Y" and saffl="Y" and (vax101dt>. or vax102dt>.) and not (not
    missing(efficacy) and not missing(siteexcl)) then
    AA1EFFL="Y";
else
    AA1EFFL="N";

if randfl="Y" and saffl="Y" and vax101dt>. and
    vax102dt>. and (UNBLNDDT=. or (UNBLNDDT>=vax102dt+7>.) and not (not
    missing(efficacy) and not missing(siteexcl)) then
    AA2EFFL="Y";
else
    AA2EFFL="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="Phase 3 3000 Subjects Flag"
    DS30KFL="Phase 3 30k Subjects Flag"
    OPBOUFL="Subjects Received Placebo & unblinded" JPNFL="Japanese Subject Flag"
    MULENRFL="Multiply Enrolled Subjects"
    PEDREAFFL="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 actarm="BNT162b2 Phase 2/3 (30 mcg)" and
    vax101="BNT162b2 (30 (*ESC*){unicode 03BC}g)" and
    vax102="BNT162b2 (30 (*ESC*){unicode 03BC}g)" 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 not missing(multiple) then
    MULENRFL="Y";

if reactofl="Y" and phasen ne 1 and agegr4n in (1 2) then
```



090177e196b7a9f0\Final\Final On: 06-Apr-2021 21:45 (GMT)

```
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.xlsx"
out=pop12_25 dbms=xlsx replace;

getnames=yes;

run;*/

*Check file name before finalization;

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;

proc sql;

create table __col as
```

```

select * from dataprot.co

where strip(uppercase(coref)) = 'SAMPLE COLLECTED' and strip(visit) = 'V3_MONTH1_POSTVAX2_L'
and coval = 'Y' and codtc ^= ''

order by usubjid, codtc;

quit;

data __co2;

set __col;

by usubjid codtc;

if first.usubjid;

run;

** Get IS data. **;

data __is(keep = usubjid testcd test cat stresc adt visitnum visit spec method);

set is_rep;

where strip(istestcd) in ('C19NIG');

length testcd $8 test $40 cat stresc spec method $200;

testcd = strip(istestcd);

test = strip(istest);

cat = strip(iscat);

stresc = uppercase(strip(isstresc));

spec = strip(isspec);

method = strip(ismethod);

adt = input(isdtc, ?? yymmdd10.);

format adt yymmdd10.;

run;

** Get MB data. **;

data __mb(keep = usubjid testcd test cat stresc adt visitnum visit_ mbloc spec method
rename = (visit_ = visit));

set dataprot.mb;

where (uppercase(strip(mbtest)) = 'SEVERE ACUTE RESP SYNDROME CORONAVIRUS 2' and
uppercase(strip(mbmetho)) = "IMMUNOCHROMATOGRAPHY") or

uppercase(strip(mbtest)) in ('CEPHEID RT-PCR ASSAY FOR SARS-COV-2', 'CEPHEID RT-PCR ASSAY OF
SARS-COV-2') and

uppercase(strip(mbmetho)) = 'REVERSE TRANSCRIPTASE PCR');

length testcd $8 test $40 cat stresc spec method $200 visit_ $64;

if uppercase(strip(mbtest)) = 'SEVERE ACUTE RESP SYNDROME CORONAVIRUS 2' and strip(spdevid)

```

090177e196b7a9f9Final\Final On: 06-Apr-2021 21:45 (GMT)

```

not in ('34','44','68') then do;

mborres = 'UNKNOWN';

mbstresc = 'UNK';

end;

testcd = strip(mbtestcd);

test = strip(mbtest);

cat = strip(mbcatt);

stresc = upcase(strip(mbstresc));

visit_ = strip(visit);

spec = strip(mbspec);

method = strip(mbmmethod);

adt = input(mbdtc, ?? yymmdd10.);

format adt yymmdd10.;

run;

proc sort data = __mb out = __mb1 nodup;

by usubjid testcd adt visitnum stresc;

run;

data __rslt1;

set __is __mb1;

if strip(stresc) = 'INDETERMINATE' then stresc = 'IND';

else if strip(stresc) = 'UNKNOWN' then stresc = 'UNK';

else if strip(stresc) in ('NEGATIVE','NEG') then stresc = 'NEG';

else if strip(stresc) in ('POSITIVE','POS') then stresc = 'POS';

if strip(testcd) in ('C19NIG') then grp = 21;

else if strip(testcd) in ('RTCOV2NS') then grp = 22;

else if strip(testcd) in ('SARSCOV2') then grp = 23;

if stresc = '' then stat = 0;

else stat = input(put(stresc,$stat.), ?? best.);

run;

proc sql;

create table __rslt2 as

select * from __rslt1 left join (select vax101dt, vax102dt, BLDV6DT, phasen, phase from

ds1 as b) on

strip(usubjid) = strip(b.usubjid);

```

```

create table __rslt2a as
select * from __rslt2 left join (select codtc from __co2 as b) on
strip(usubjid) = strip(b.usubjid)
order by usubjid, vaxl01dt, vaxl02dt, visitnum, visit, grp, adt;
quit;

data __rslt3(drop = codtc)
__rslt3_flags(keep = usubjid vaxl01dt vaxl02dt vldrslfl vrblngfl vrv3ngfl crdlngfl
crd2ngfl pdpl7fl_ pdp27fl_);

set __rslt2a;

by usubjid vaxl01dt vaxl02dt visitnum visit grp adt;

  ** Derive result flags. **;

if first.usubjid then do;

vrblngfl = 'U';

vrv3ngfl = 'U';

crdlngfl = 'U';

crd2ngfl = 'U';

pdpl7fl_ = 'N';

pdp27fl_ = 'N';

and;

vldrslfl = 'N';

if strip(visit) = 'V1_DAY1_VAX1_L' then do;
if . < adt <= vaxl01dt then vldrslfl = 'Y';
if vldrslfl = 'Y' and strip(put(stat,stat.)) = 'POS' then do;
if grp = 21 then vrblngfl = 'N';
if grp = 22 then crdlngfl = 'N';
and;
if vldrslfl = 'Y' and strip(put(stat,stat.)) = 'NEG' then do;
if grp = 21 then vrblngfl = 'Y';
if grp = 22 then crdlngfl = 'Y';
and;
if last.visitnum and vrblngfl = 'Y' and crdlngfl = 'Y' then pdpl7fl_ = 'Y';
and;

else if strip(visit) = 'V2_VAX2_L' then do;
if . < adt <= vaxl02dt then vldrslfl = 'Y';

```

090177e196b7a9f9\Final\Final On: 06-Apr-2021 21:45 (GMT)

000177e196b79af0\Final\Final on: 06-Apr-2021 21:45 (GMT)

```
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.visitnum and vrblngfl = 'Y' and crdlnngfl = 'Y' and crd2ngfl = 'Y' then pdp27fl_ =
'Y';
end;

else if strip(visit) not in ('V1_DAY1_VAX1_L','V2_VAX2_L') and grp = 21 then do;
if vax102dt < adt <= COALESCE(BLDV6DT,vax102dt+28) then vldrslfl = 'Y';
if vldrslfl = 'Y' and strip(put(stat,stat.)) = 'POS' and grp = 21 then vrv3ngfl = 'N';
if vldrslfl = 'Y' and strip(put(stat,stat.)) = 'NEG' and grp = 21 then vrv3ngfl = 'Y';
end;

else if strip(visit) not in ('V1_DAY1_VAX1_L','V2_VAX2_L','V3_MONTH1_POSTVAX2_L') and grp
^= 21 then do;

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;

codt = input(codtc, ?? yymmdd10.);
format codt yymmdd10.;
output __rslt3;

if last.usubjid then output __rslt3_flags;

retain vrblngfl vrv3ngfl crdlnngfl crd2ngfl pdp17fl_ pdp27fl_;

run;

proc sort data = __rslt3 out = __rslt4(drop = cat spec method mbloc);
by usubjid vax101dt vax102dt visitnum visit grp stat adt;

where cncrslfl = 'Y';

run;

data __rslt5 __rslt5a(keep = usubjid vax101dt vax102dt BLDV6DT codt phasen adt stat
vrblngfl rename = (adt = nva_bl_dt stat = nva_bl)) __rslt5b(keep = usubjid vax101dt
vax102dt BLDV6DT codt phasen adt stat vrv3ngfl rename = (adt = nva_v3_dt stat = nva_v3))
__rslt5c(keep = usubjid vax101dt vax102dt BLDV6DT codt phasen adt stat crdlnngfl rename =
(adt = cnt_ldt stat = cnt_1)) __rslt5d(keep = usubjid vax101dt vax102dt BLDV6DT codt
phasen adt stat crd2ngfl rename = (adt = cnt_2dt stat = cnt_2)) __rslt5e(keep = usubjid
vax101dt vax102dt BLDV6DT codt phasen visitnum visit adt stat vldrslfl rename = (vldrslfl
c_vldrslfl adt = cnt_unp_dt stat = cnt_unp)) __rslt5f(keep = usubjid vax101dt vax102dt
BLDV6DT codt phasen visitnum visit adt stat vldrslfl rename = (vldrslfl = l_vldrslfl adt
```

```

= lcl_unp_dt stat = lcl_unp));
set __rslt4;
by usubjid vax101dt vax102dt visitnum visit grp stat adt;
if last.grp then keepflg = 1;
output __rslt5;
if keepflg = 1 then do;
if grp = 21 and strip(visit) = 'V1_DAY1_VAX1_L' then output __rslt5a;
if grp = 21 and strip(visit) ^= 'V1_DAY1_VAX1_L' and vldrslfl = 'Y' then output __rslt5b;
if grp = 22 and strip(visit) = 'V1_DAY1_VAX1_L' then output __rslt5c;
if grp = 22 and strip(visit) = 'V2_VAX2_L' then output __rslt5d;
if grp = 22 and strip(visit) not in ('V1_DAY1_VAX1_L','V2_VAX2_L') then output __rslt5e;
if grp = 23 and strip(visit) not in ('V1_DAY1_VAX1_L','V2_VAX2_L') then output __rslt5f;
end;
run;
data __rslt6;
merge __rslt5a(in = a) __rslt5b(in = b) __rslt5c(in = c) __rslt5d(in = d);
by usubjid vax101dt vax102dt;
if first.usubjid and last.usubjid then dupflg = 0;
else dupflg = 1;
run;
data __cnt_lcl1;
merge __rslt5e(in = a) __rslt5f(in = b);
by usubjid vax101dt vax102dt visitnum visit;
** Process central and local lab rerults. **;
** Conclude NAAT result for unplanned visits. **;
if cnt_unp ^= . and cnt_unp_dt ^= . then do;
naat_unp = cnt_unp;
naat_unp_dt = cnt_unp_dt;
end;
else if lcl_unp ^= . and lcl_unp_dt then do;
naat_unp = lcl_unp;
naat_unp_dt = lcl_unp_dt;
end;
format naat_unp_dt yymmdd10.;

```

000177e496b7a9f0Final On: 06-Apr-2021 14:45 (GMT)

```

run;

proc sort data = __cnt_lcl1 out = __cnt_lcl2;

by usubjid vax101dt vax102dt descending naat_unp naat_unp_dt;

where naat_unp ^= 2;

run;

data __cnt_lcl2;

set __cnt_lcl2;

by usubjid vax101dt vax102dt descending naat_unp naat_unp_dt;

if first.usubjid then keepflg = 1;

run;

data __rslt7;

merge __rslt6(in = a drop = dupflg) __cnt_lcl2(in = b keep = usubjid vax101dt vax102dt
BLDV6DT phasen naat_unp naat_unp_dt keepflg where = (keepflg = 1));

by usubjid vax101dt vax102dt;

if a and not b then mflg = 1;

if a and b then mflg = 2;

if not a and b then mflg = 3;

run;

* (b) (6) 26Feb2021 to include all subjects with any record;

proc sort data=__rslt4 out=__rslt4_1 nodupkey;

by usubjid;

run;

data __rslt8;

merge __rslt4_1(in=a) __rslt7(in=b);

by usubjid vax101dt vax102dt;

if a or b;

if nva_bl = 4 or cnt_1 = 4 or cnt_2 = 4 or nva_v3 = 4 then pclmd2fl = 'Y';

else pclmd2fl = 'N';

if naat_unp = 4 and ((. < naat_unp_dt <= nva_v3_dt) or (nva_v3_dt = . and . < naat_unp_dt
= codt) or (nva_v3_dt = . and codt = . and . < naat_unp_dt <= BLDV6DT) or (nva_v3_dt = .
and codt = . and BLDV6DT=. and . < naat_unp_dt <= sum(vax102dt,28)) or (nva_v3_dt = . and
codt = . and BLDV6DT=. and vax102dt=. and . < naat_unp_dt <= sum(vax101dt,28))) then
pclmd2fl = 'Y';

label pclmd2fl = 'Positive SARS-CoV-2 Prior to 1MP Dose 2';

run;

data __rslt9(keep = usubjid pclmd2fl);

```

```

set __rslt8;

by usubjid vaxl01dt vaxl02dt;

where phasen >= 2;

proc sort;

by usubjid descending pclmd2fl;

run;

data __rslt10;

set __rslt9;

by usubjid descending pclmd2fl;

if first.usubjid;

run;

proc sort data=dataprot.mh out=_mh(keep=usubjid) nodupkey;

by usubjid;

where mhdecod in ('COVID-19' 'SARS-CoV-2 antibody test positive');

run;

data __rslt11;

merge __rslt10 _mh(in=a);

by usubjid;

if a then pclmd2fl='Y';

run;

data adsl;

merge adsl(in = a) __rslt11(in = b);

by usubjid;

if vaxl01dt=. or vaxl02dt=. then pclmd2fl='';

run;

*/

*proc import datafile="&expath./hiv-preferred-terms.xlsx" out=hivpt dbms=xlsx

replace;

*EXLX;

getnames=yes;

run;*/

*Check file name before finalization;

proc import datafile="&expath./201114 HIV preferred terms.xlsx" out=hivpt

```

090177e4966b7a9f0\Final On: 06-Apr-2021 14:45 (GMT)



090177e196b7a9f0\Final\Final On: 06-Apr-2021 21:45 (GMT)

```
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;

    ***** (b) (6) (14Nov2020) - Flag for HIV +ve Subjects *****;

    if a and b then

        HIVFL="Y";

    else

        HIVFL="N";

    label HIVFL="HIV Positive Subjects Flag";

    ***** (b) (6) (14Nov2020) - Set all Efficacy Flags to N for Phase 1 subjects *****;
```

```

if phasen eq 1 then
    do;
        EVALEFFL="N";
        AAI1EFFL="N";
        AAI2EFFL="N";
    end;
run;

/**** START - Setting up ADSYMPT dataset *****/;

** Get FA data. **;

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

data fac1 face_stdtd(keep=usubjid faorres visitnum visit
    rename=(faorres=fastdtdtc)) face_endtd(keep=usubjid faorres visitnum visit
    rename=(faorres=faendtdtc)) face_ong(keep=usubjid faorres visitnum visit
    rename=(faorres=faong));
set face;
by usubjid visitnum visit fatestd faobj faorres;

if upcase(strip(fatestd))='FSYMDATE' then
    output face_stdtd;
else if upcase(strip(fatestd))='LSYMDATE' then
    output face_endtd;
else if upcase(strip(fatestd))='SYMONGO' then
    output face_ong;
else
    output fac1;
run;

data face2;

```

000177e196b7a9f0\Final\Final On: 06-Apr-2021 21:45 (GMT)

```
merge facel(in=a) face_stdtd(in=b) face_endtd(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 aendtd 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='NLTSTSMML';
    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;
    end;
    aval=.;
    adt=input(fadtc, ?? yymmdd10.);
    astdt=input(fastdte, ?? 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;
    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 is_rep;

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(isdtt, ?? 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;

```

090177e196b7a9f0\Final\Final On: 06-Apr-2024 21:45 (GMT)

```
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(mbcatt);
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(mbdttc, ?? 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 mbl;

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 fatestd fatest

    faobj facat fascat faorres fastresc fadrvfl visitnum visit fadtc) out=__face;

```

090177e196b7a9f0\Final\Final On: 06-Apr-2021 21:45 (GMT)

```

by usubjid visitnum visit fatestdc faobj faorres;

where upcase(strip(facat))='EFFICACY';

run;

data __face1 __face_stdtd(keep=usubjid faorres visitnum visit
    rename=(faorres=fastdtd)) __face_endtd(keep=usubjid faorres visitnum visit
    rename=(faorres=faendtd)) __face_ong(keep=usubjid faorres visitnum visit
    rename=(faorres=faong));

set __face;

by usubjid visitnum visit fatestdc faobj faorres;

if upcase(strip(fatestdc))='FSYMDATE' then
    output __face_stdtd;
else if upcase(strip(fatestdc))='LSYMDATE' then
    output __face_endtd;
else if upcase(strip(fatestdc))='SYMONGO' then
    output __face_ong;
else
    output __face1;

run;

data __fa(keep=usubjid domain adt astdt aendt visitnum visit);

merge __face1(in=a) __face_stdtd(in=b) __face_endtd(in=c) __face_ong(in=d);

by usubjid visitnum visit;

if a;

adt=input(fadtd, ?? yymmdd10.);

astdt=input(fastdtd, ?? yymmdd10.);

aendt=input(faendtd, ?? yymmdd10.);

format adt astdt aendt yymmdd10.;

run;

* Get data from HO. **;

proc sql;

create table __ho1 as select * from dataprot.ho left join

```

090177e196b7a9f0\Final\Final On: 06-Apr-2021 21:45 (GMT)

0900177e196b7a9f0\Final\Final On: 06-Apr-2021 21:45 (GMT)

```
(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 is_rep;
```

```
where strip(istestcd) in ('C19NIG');

adt=input(isdte, ?? 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(lbdte, ?? 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(mbmeth))='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(mbmeth))='REVERSE TRANSCRIPTASE PCR');

length visit_ $64;

visit_=strip(visit);

adt=input(mbdte, ?? yymmdd10.);

astdt=.;
```

090177e496b7a9f0\Final\Final\_On: 06-Apr-2021 21:45 (GMT)

```
aendt=.;
format adt astdt aendt yymmdd10.;

run;

** Get PR data. **;

data __pr(keep=usubjid domain adt astdt aendt visitnum visit);
    set dataprot.pr;
    where strip(prcat)='GENERAL NON-DRUG TREATMENT' and prtrt ^='';
    adt=input(prdtc, ?? yymmdd10.);
    astdt=input(prstdtc, ?? yymmdd10.);
    aendt=input(prendtc, ?? yymmdd10.);
    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 __pr __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 aenddt 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 aenddt 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 aenddt 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;

```

090177e196b7a9f0\Final\Final On 06-Apr-2021 21:45 (GMT)



```

** 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

```

090177e196b7a9f0\Final\On: 06-Apr-2021 21:45 (GMT)

```

        (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_;


```

0901777e196b7a9f0\Final\Final On: 06-Apr-2024 21:45 (GMT)

```
                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

```

090177e196b7a9f0\Final\Final On: 06-Apr-2021 21:45 (GMT)

090177e196b7a9f0\Final\Final On: 06-Apr-2021 21:45 (GMT)

```
        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 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 ne 1 and 12 <=aget01 <=25 and EVAL02FL='Y') 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;

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 asdt;

run;

data symp_all_1 ord_data_1(keep=usubjid visitnum visit avisitn avisit srtedt)

    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 asdt aendt);

by usubjid avisitn paramn aval avalc adt asdt;

stat=input(put(avalc, $stat.), ?? best.);

srtedt=asdt;

** 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)='V1_DAY1_VAX1_L' then

    do;

        grp=21;

        ** These number assignments are used below. **;

        srtedt=adt;

        output symp_all_1;

```

090177e196b7a9f0\Final\Final On: 06-Apr-2021 21:45 (GMT)

```
end;

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

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

if strip(paramcd) in ('C19NIG') and strip(avisit) ^='V1_DAY1_VAX1_L' then
  do;
    grp=24;
    srtedt=adtd;
    output symp_all_1;
  end;

if grp ^=. then
  output ord_data_1;

if visitnum ^=avisitn or visit ^=avisit then
  output vis_colsp1;

format adtd astdtd aendtd srtedt yymmdd10.;

run;

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

data ord_data_1b;
```

```
set ord_data_1a;
by usubjid srttdt 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, srttdt;
quit;

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

if first.usubjid then
```

```
srtord_b=0;

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

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

    if strip(avisit)='V2_VAX2_L' 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_;
    end;
  end;

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 srtedt=b.srtedt;

  ** 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 ads1 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_stdtd grp_endtd visitnum_
  visit_ rename=(grp_stdtd=astdt grp_endtd=aendt visitnum_=visitnum
  visit_=visit)) nva_naatl(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_stdtd=astdt;
    grp_endtd=aendt;
    grp_stat=stat;
  end;

```

090177e196b7a9f0\Final\Final On: 06-Apr-2021 21:45 (GMT)

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

if grp_stat < stat or grp in (7, 8) then
    do;
        grp_stdtd=astdt;
        grp_endtd=aendtd;
        grp_stat=stat;
        visitnum_=visitnum;
        visit_=visit;
    end;

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

if (. < grp_endtd < aendtd) or aendtd=. then
    grp_endtd=aendtd;

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_naatl;

format grp_stdtd grp_endtd yymmdd10.;
retain grp_stdtd grp_endtd grp_stat visitnum_ visit_;
run;

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

data symp2(drop=vis_endtfl setflg vis_stat vis_astdt vis_aendtd vis_endtfl_cdc
```

```
        setflg_cdc vis_stat_cdc vis_astdt_cdc vis_aendt_cdc);  
set sympl;  
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_naatl as select * from nva_naatl left join

```

090177e196b7a9f0\Final\Final On: 06-Apr-2021 21:45 (GMT)

```

        (select usubjid as usubjid_v, min(astdt) as astdt_sym_v format
yymmdd10., max(aenddt) as aenddt_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_naatl1b as select * from nva_naatl1a left join

        (select usubjid as usubjid_av, min(astdt) as astdt_sym_av format
yymmdd10., max(aenddt) as aenddt_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_naatl2 nva_naatl_flags(keep=usubjid vax101dt vax102dt dthdt vldrs1fl
        vrblngfl crd1ngfl crd2ngfl pdp17fl_ pdp27fl_);

**** Use this dataset for flags ****;

set nva_naatl1b;

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;
vldrs1fl='N';

if strip(avisit)='V1_DAY1_VAX1_L' 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
            crdlngfl='N';

    end;

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

        if grp=21 then
            vrblngfl='Y';

        if grp=22 then
            crdlngfl='Y';

    end;

if last.avisitn and vrblngfl='Y' and crdlngfl='Y' then
    pdp17fl_='Y';

end;

else if strip(avisit)='V2_VAX2_L' 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';

    end;
```

```

        if last.avisitn and vrblngfl='Y' and crdlnngfl='Y' and crd2ngfl='Y' then
            pdp27fl_='Y';
        end;
    else if strip(avisit) not in ('V1_DAY1_VAX1_L', 'V2_VAX2_L') 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_naata2;

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

proc sort data=nva_naata2 out=nva_naata3(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_naata_flags where
        VRBLNGFL='Y' and CRD1NGFL='Y' and CRD2NGFL='Y' and usubjid in (select
            distinct usubjid from adsl where saffl='Y');
quit;

data nva_naata4 nva_naata4a(keep=usubjid adt stat rename=(adt=nva_dt stat=nva))
    nva_naata4b(keep=usubjid adt stat rename=(adt=cnt_1dt stat=cnt_1))
    nva_naata4c(keep=usubjid adt stat rename=(adt=cnt_2dt stat=cnt_2))
    nva_naata4d(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_naata4e(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_naata4f(keep=recseq usubjid vax101dt vax102dt adt grp stat
    rename=(recseq=recseq_f adt=nva_v3_dt stat=nva_v3)) nva_naata4g(keep=recseq
    usubjid vax101dt vax102dt adt grp stat cl9cnv_dy rename=(recseq=recseq_g
    adt=nva_cnv_dt stat=nva_cnv));
set nva_naata3;

```

```
by usubjid vax101dt vax102dt avisitn avisit visitnum visit grp stat adt;

if grp=24 and strip(avisit) ^='V3_MONTH1_POSTVAX2_L' 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_naata4;

if keepflg=1 then
  do;

    if grp=21 and strip(avisit)='V1_DAY1_VAX1_L' then
      output nva_naata4a;

    if grp=22 and strip(avisit)='V1_DAY1_VAX1_L' then
      output nva_naata4b;

    if grp=22 and strip(avisit)='V2_VAX2_L' then
      output nva_naata4c;

    if grp=22 and strip(avisit) not in ('V1_DAY1_VAX1_L', 'V2_VAX2_L') then
      output nva_naata4d;

    if grp=23 and strip(avisit) not in ('V1_DAY1_VAX1_L', 'V2_VAX2_L') then
      output nva_naata4e;

    if grp=24 then
      do;

        if strip(avisit)='V3_MONTH1_POSTVAX2_L' then
          output nva_naata4f;
        else
          output nva_naata4g;
```

```
        end;

    end;

run;

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

data cnt_lcl1;

    merge nva_naata4d(in=a) nva_naata4e(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;
```

090177e196b7a9f9\Final\Final On: 06-Apr-2021 21:45 (GMT)

```
        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 vldrslf1)
        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;
        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 vaxl01dt vaxl02dt 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 vaxl01dt vaxl02dt dthdt
      pdsymfl_ pdsdmfl_ cdcsymfl_ sevsymfl_ sevcdcf_ pdrmufl_ pdrmupfl_ cdcrmufl_
      cdrmupfl_ pdp1fl_ pdp17fl_ pdp27fl_ pdp214fl_ cdp1fl_ cdp17fl_ cdp27fl_
      cdp214fl_);

set symp3d end=eof;

by usubjid vaxl01dt vaxl02dt srtord avisitn avisit grpcat grp astdt;

** Setting the flags. **;

if first.usubjid then
  do;
    pdsymfl_='N';
    pdsdmfl_='N';
    cdcsymfl_='N';
    sevsymfl_='N';
    sevcdcf_='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_='';
  end;

```

```

        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;
        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;
                pdrmufll_='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;
      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;
                cdcrmuf1_='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
        sevcdcf1_='Y';

    if dthdt ^=. then
        do;
            sevsymfl_='Y';
            sevcdcf1_='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 stdyl < 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
                    pdplfl_='N';

                if grp=2 then
                    cdplfl_='N';
            end;
    end;

```

```

        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;
                        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;
            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;

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;

```

```

        pdp1f1_='N';
    end;

    if vax101dt ^=. and vax101dt <=tmp_unp_dt < sum(vax101dt, 7) then
        do;
            pdp17f1_='N';
            pdp27f1_='N';
            cdp17f1_='N';
            cdp27f1_='N';
        end;

        if vax101dt ^=. and vax102dt ^=. and vax101dt <=tmp_unp_dt < sum(vax102dt,
            7) then
            do;
                pdp27f1_='N';
                cdp27f1_='N';
            end;

            if vax101dt ^=. and vax102dt ^=. and vax101dt <=tmp_unp_dt < sum(vax102dt,
                14) then
                do;
                    pdp214f1_='N';
                    cdp214f1_='N';
                end;
            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_ sevcdcf1_
        pdrmufl_ pdrmupfl_ cdcrmufl_ cdrmupfl_ filocrfl_pd_ filocrfl_cdc_
        filocrfl_sev_ filocrfl_sev_cdc_ pdp1f1_ pdp17f1_ pdp27f1_ pdp214f1_ cdp1f1_

```

```

cdp17f1_ cdp27f1_ cdp214f1_ 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), Convalescent 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 ('V3_MONTH1_POSTVAX2_L') and not missing(SVSTDTC))) a inner join
        adsl (where=(phasen ne 1 and 12 <=aget01 <=25)) b on a.usubjid=b.usubjid
        order by usubjid;

    create table sv_conv 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 ne 1 and 12 <=aget01 <=25)) b on a.usubjid=b.usubjid
        order by usubjid;

    create table sv_V101 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
        ne 1 and 12 <=aget01 <=25)) b on a.usubjid=b.usubjid order by usubjid;

    create table sv_V201 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 ne 1 and 12 <=aget01 <=25)) b on a.usubjid=b.usubjid
        order by usubjid;

    create table dt0 as select a.usubjid, a.vax101dt, a.vax102dt, 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 ne 1 and 12 <=aget01 <=25)) a
left join sv_v3 b on a.usubjid=b.usubjid left join sv_conv c on
a.usubjid=c.usubjid left join sv_V101 d on a.usubjid=d.usubjid left join
sv_V201 e on a.usubjid=e.usubjid order by usubjid;

```

```
quit;
```

\*\*\*\*\* Identify subjects with C19NIG results for Visit 3 (V3\_MONTH1\_POSTVAX2\_L),  
Convalscnt Visits (A1,B1,C1,...), V101 and V201 visits from ADSYMPT domains \*\*\*\*\*;

```
proc sql noprint;
```

```

create table c19_v3 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 ne 1 and 12 <=aget01 <=25)) b on a.usubjid=b.usubjid
order by usubjid;

```

```

create table c19_conv 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 ne 1 and
12 <=aget01 <=25)) b on a.usubjid=b.usubjid order by usubjid;

```

```

create table c19_v101 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 ne 1 and
12 <=aget01 <=25)) b on a.usubjid=b.usubjid order by usubjid;

```

```

create table c19_v201 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 ne 1 and 12 <=aget01 <=25)) b on a.usubjid=b.usubjid
order by usubjid;

```

```
quit;
```

```
data c19;
```

```

merge c19_v3 (in=a) c19_conv (in=b) c19_v101 (in=c) c19_v201 (in=d);
by usubjid;

```

```
if a or b or c or d;
```



```

run;

***** Identify subjects with RTCOV2NS results for Visit 3 (V3_MONTH1_POSTVAX2_L),
Convalscient Visits (A1,B1,C1,...), V101 and V201 visits from ADSYMPT domains *****;

proc sql noprint;

    create table rt_v3 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 ne 1 and
        12 <=aget01 <=25)) b on a.usubjid=b.usubjid order by usubjid;

    create table rt_conv 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 ne 1 and
        12 <=aget01 <=25)) b on a.usubjid=b.usubjid order by usubjid;

    create table rt_v101 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 ne 1 and
        12 <=aget01 <=25)) b on a.usubjid=b.usubjid order by usubjid;

    create table rt_v201 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 ne 1 and 12 <=aget01 <=25)) b on a.usubjid=b.usubjid
        order by usubjid;

quit;

data rt;

    merge rt_v3 (in=a) rt_conv (in=b) rt_v101 (in=c) rt_v201 (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 miss_vis3dt nomiss_vis3dt;

    merge dt0 (in=a) c19 rt;

```

090177e196b7a9f0\Final\On: 06-Apr-2021 21:45 (GMT)

```
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(c19vall1) or not missing(rtvall1)) 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;
```

```
if missing(vis3dt) then
    output miss_vis3dt;

if not missing(vis3dt) then
    output nomiss_vis3dt;

run;

proc sort data=nomiss_vis3dt;
    by usubjid vis3dt;
run;

data nomiss_vis3dt;
    set nomiss_vis3dt;
    by usubjid vis3dt;

    if first.usubjid;
run;

***** Without Results - Only Visit - Date Get Visit 3 Date cut off *****;

data dt_c19_rt1 miss_vis3dt1 nomiss_vis3dt1;
    merge dt0 (in=a) c19 rt;
    by usubjid;

    if a;
    format vis3dt date9.;

    if not missing(v3dt) then
        vis3dt=v3dt;
    else
        do;

            if not missing(convdt) then
                do;

                    if 28 <=convdy <=42 or (not missing(v3dt) and
                        v3dt - 7 <=convdt <=v3dt + 7) then
                        vis3dt=convdt;
```

090477e196b7a9f0\\Final On 06-Apr-2021 21:45 (GMT)

```
end;

if 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(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_rt1;

if missing(vis3dt) then
  output miss_vis3dt1;

if not missing(vis3dt) then
  output nomiss_vis3dt1;

run;

proc sort data=nomiss_vis3dt1;
  by usubjid vis3dt;
run;

data nomiss_vis3dt1;
  set nomiss_vis3dt1;
  by usubjid vis3dt;

  if first.usubjid;
run;
```

\*\*\*\*\* Combine VRBLNGFL, CRD1NGFL, CRD2NGFL and VIS3DT to check unique subjects \*\*\*\*\*;

proc sql noprint;

```
create table nva_naats_vis3 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_naats_flags a left
join nomiss_vis3dt b on a.usubjid=b.usubjid order by usubjid;
```

quit;

proc sql noprint;

```
select count (distinct usubjid) into :nl from nva_naats_vis3 where EV1MD2FL='Y'
and usubjid in (select distinct usubjid from adsl where saffl='Y');
```

quit;

%put &nl.;

\*\*\*\*\* Subjects with POS swabs after concluded results \*\*\*\*\*;

proc sql noprint;

```
create table pos_s as select distinct a.usubjid, a.naat_unp, a.naat_unp_dt,
b.vis3dt, b.vax101dt, b.vax102dt from cnt_lcl2 (where=(naat_unp ne 2)) a left
join nomiss_vis3dt 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 as select distinct a.usubjid, a.cnt_unp, a.cnt_unp_dt,
b.vis3dt, b.vax101dt, b.vax102dt from cnt_lcl2 (where=(cnt_unp ne 2)) a left
join nomiss_vis3dt b on a.usubjid=b.usubjid
where . < b.vax101dt <=a.cnt_unp_dt <=b.vis3dt order by usubjid, cnt_unp_dt;
```

quit;

090177e196b7a9f0\Final\Final On: 06-Apr-2021 21:45 (GMT)

```

proc sort nodupkey;
    by usubjid;
run;

***** Subjects with Symptoms but no valid NEG result *****;

proc sql noprint;
    create table sym_n as select distinct a.usubjid, a.astdt, a.stat, a.naat_unp,
        a.naat_unp_dt, b.vis3dt, b.vaxl01dt, b.vaxl02dt from symp4 (where=((not
        missing(astdt) or missing(astdt)) and stat=4 and naat_unp ne 2)) a left join
        nomiss_vis3dt b on a.usubjid=b.usubjid
        where . < b.vaxl01dt <=a.astdt <=b.vis3dt order by usubjid, naat_unp_dt;
quit;

proc sort nodupkey;
    by usubjid;
run;

***** Subjects with POS N-binding Assay *****;

proc sql noprint;
    create table pos_n as select distinct a.usubjid, a.avalc, a.adt, a.avisit,
        b.vis3dt, b.vaxl01dt, b.vaxl02dt from adsympt (where=(paramcd in ('C19NIG')
        and avisit not in ('V1_DAY1_VAX1_L') and avalc not in ('NEG'))) a left join
        nomiss_vis3dt b on a.usubjid=b.usubjid
        where . < b.vaxl01dt <=a.adt <=b.vis3dt order by usubjid, adt;
quit;

proc sort nodupkey;
    by usubjid;
run;

***** Derive the final EV1MD2FL flag *****;

proc sort data=adsl out=saf_pop (keep=usubjid UNBLNDDT) nodupkey;
    by usubjid;
    where EVAL02FL='Y' and phasen ne 1 and 12 <=aget01 <=25;

```

090177e196b7a9f0\Final\Qn: 06-Apr-2021 21:45(GMT)

```

run;

data ev1;

    merge saf_pop (in=a) nva_naot_vis3 (in=b) pos_s_c (in=c) sym_n (in=d)
        pos_n (in=e) pos_s (in=f);
    by usubjid;

    if a;

    if (b and c) or (b and d) or (b and e) or (b and f) then
        EV1MD2FL='N';

    if missing(EV1MD2FL) then
        EV1MD2FL='N';

    **** (b) (6) (22MAR2021): As pe (b) (6) we need to include subjects who UNBLNDDT before
    Visit 3 Date *****;

    **** From (b) (6) (22Mar2021, 1:01pm in (b) (6)): unlike AE and efficacy
    which could be *****;

    **** affected by the bias of unblinding, immuno data is objective, impact of
    different bahavior *****;

    **** after unblinding. such as natural infection, will be captured by 'without
    evidence of infection' **;

    **** condition, but the without evidence of infection flag does not need to consider
    unblinding. *****;

    /* if . < UNBLNDDT < vis3dt then EV1MD2FL = "N"; */;

run;

proc sort out=_ev1 (keep=usubjid EV1MD2FL);
    by usubjid EV1MD2FL;
run;

proc freq data=_ev1;
    table EV1MD2FL / list;
run;

**** Merging EV1MD2FL with ADSL *****;

data adsl;

    merge adsl (in=a) _ev1;

```

090177e196b7a9f0\Final\On: 06-Apr-2021 21:45 (GMT)

```

by usubjid;

if a;

if missing(EV1MD2FL) then
    EV1MD2FL='N';

run;

*****
;
* Specification 9 *;
* FOLLOW UP CATEGORIES *;
* 1 - Censor Date. *;
* 2 - Follow up variables in days. *;
* 3 - Follow up categories. *;
*****
;
* (b) (6) 23Feb2021 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.;

    if randfl="Y" then
        do;
            STCSRDT=min(eosdcdt, "&cutoff2"d);

            if (tr02sdt>. or UNBLNDDT>.) and boostfl ne "Y" 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;
    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="&cutoff2"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>.) and boostfl ne "Y" 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>.) and boostfl ne "Y" 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;

proc sql;
  select floor(max(FUP2CUT/28)/1)+1 into: maxloop from adsl where FUP2CUT>.;
quit;

data adsl;
```

```
set adsl;  
length FUP2CAT1 $20;  
  
if randfl="Y" then  
  do;  
  
    if 0<=FUP2CUT/28<1 then  
      do;  
        FUP2CAT1="0-1 month";  
        FUP2CA1N=1;  
      end;  
    else if 1<=FUP2CUT/28<2 then  
      do;  
        FUP2CAT1="1-2 months";  
        FUP2CA1N=2;  
      end;  
    else if 2<=FUP2CUT/28<3 then  
      do;  
        FUP2CAT1="2-3 months";  
        FUP2CA1N=3;  
      end;  
    else if 3<=FUP2CUT/28<4 then  
      do;  
        FUP2CAT1="3-4 months";  
        FUP2CA1N=4;  
      end;  
    else if 4<=FUP2CUT/28<5 then  
      do;  
        FUP2CAT1="4-5 months";  
        FUP2CA1N=5;  
      end;  
    else if 5<=FUP2CUT/28<6 then  
      do;  
        FUP2CAT1="5-6 months";
```

```

        FUP2CA1N=6;

    end;

else if 6<=FUP2CUT/28<7 then

    do;

        FUP2CAT1="6-7 months";

        FUP2CA1N=7;

    end;

else if 7<=FUP2CUT/28<8 then

    do;

        FUP2CAT1="7-8 months";

        FUP2CA1N=8;

    end;

else if 8<=FUP2CUT/28<9 then

    do;

        FUP2CAT1="8-9 months";

        FUP2CA1N=9;

    end;

else if 9<=FUP2CUT/28<10 then

    do;

        FUP2CAT1="9-10 months";

        FUP2CA1N=10;

    end;

else if 10<=FUP2CUT/28<11 then

    do;

        FUP2CAT1="10-11 months";

        FUP2CA1N=11;

    end;

end;

```

```

end;

```

```

run;

```

```

proc sql;

```

```

select floor(max(FUP2UNB/28)/1)+1 into: maxloop from ads1 where FUP2UNB>.;

```

090177e196b7a9f0\Final\Final On: 06-Apr-2021 21:45 (GMT)

```
quit;

data adsl;

  set adsl;

  length FUP2CAT2 $20;

  if randfl="Y" then
    do;

      if 0<=FUP2UNB/28<1 then
        do;
          FUP2CAT2="0-1 month";
          FUP2CA2N=1;
        end;
      else if 1<=FUP2UNB/28<2 then
        do;
          FUP2CAT2="1-2 months";
          FUP2CA2N=2;
        end;
      else if 2<=FUP2UNB/28<3 then
        do;
          FUP2CAT2="2-3 months";
          FUP2CA2N=3;
        end;
      else if 3<=FUP2UNB/28<4 then
        do;
          FUP2CAT2="3-4 months";
          FUP2CA2N=4;
        end;
      else if 4<=FUP2UNB/28<5 then
        do;
          FUP2CAT2="4-5 months";
          FUP2CA2N=5;
        end;
      else if 5<=FUP2UNB/28<6 then
```

```

do;
    FUP2CAT2="5-6 months";
    FUP2CA2N=6;
end;
else if 6<=FUP2UNB/28<7 then
do;
    FUP2CAT2="6-7 months";
    FUP2CA2N=7;
end;
else if 7<=FUP2UNB/28<8 then
do;
    FUP2CAT2="7-8 months";
    FUP2CA2N=8;
end;
else if 8<=FUP2UNB/28<9 then
do;
    FUP2CAT2="8-9 months";
    FUP2CA2N=9;
end;
else if 9<=FUP2UNB/28<10 then
do;
    FUP2CAT2="9-10 months";
    FUP2CA2N=10;
end;
end;

```

```
run;
```

```
proc sql;
```

```
select floor(max(FPX1CUT/28)/1)+1 into: maxloop from adsl where FPX1CUT>.;
```

```
quit;
```

```
data adsl;
```

```
set adsl;
```

0900177e1196b7a9f0\Final\Final On: 06-Apr-2021 21:45 (GMT)



```
length FPX1CAT1 $20;

if randfl="Y" then
  do;

    if 0<=FPX1CUT/28<1 then
      do;
        FPX1CAT1="0-1 month";
        FPX1CA1N=1;
      end;
    else if 1<=FPX1CUT/28<2 then
      do;
        FPX1CAT1="1-2 months";
        FPX1CA1N=2;
      end;
    else if 2<=FPX1CUT/28<3 then
      do;
        FPX1CAT1="2-3 months";
        FPX1CA1N=3;
      end;
    else if 3<=FPX1CUT/28<4 then
      do;
        FPX1CAT1="3-4 months";
        FPX1CA1N=4;
      end;
    end;
  end;

run;

proc sql;
  select floor(max(FUP1CUT/28)/1)+1 into: maxloop from adsl where FUP1CUT>.;
quit;

data adsl;
  set adsl;
```

```
length FUP1CAT1 $20;

if randfl="Y" then
  do;

    if 0<=FUP1CUT/28<1 then
      do;
        FUP1CAT1="0-1 month";
        FUP1CA1N=1;
      end;
    else if 1<=FUP1CUT/28<2 then
      do;
        FUP1CAT1="1-2 months";
        FUP1CA1N=2;
      end;
    else if 2<=FUP1CUT/28<3 then
      do;
        FUP1CAT1="2-3 months";
        FUP1CA1N=3;
      end;
    else if 3<=FUP1CUT/28<4 then
      do;
        FUP1CAT1="3-4 months";
        FUP1CA1N=4;
      end;
    else if 4<=FUP1CUT/28<5 then
      do;
        FUP1CAT1="4-5 months";
        FUP1CA1N=5;
      end;
    else if 5<=FUP1CUT/28<6 then
      do;
        FUP1CAT1="5-6 months";
        FUP1CA1N=6;
```

```

        end;
    else if 6<=FUP1CUT/28<7 then
        do;
            FUP1CAT1="6-7 months";
            FUP1CA1N=7;
        end;
    else if 7<=FUP1CUT/28<8 then
        do;
            FUP1CAT1="7-8 months";
            FUP1CA1N=8;
        end;
    else if 8<=FUP1CUT/28<9 then
        do;
            FUP1CAT1="8-9 months";
            FUP1CA1N=9;
        end;
    else if 9<=FUP1CUT/28<10 then
        do;
            FUP1CAT1="9-10 months";
            FUP1CA1N=10;
        end;
    else if 10<=FUP1CUT/28<11 then
        do;
            FUP1CAT1="10-11 months";
            FUP1CA1N=11;
        end;
    else if 11<=FUP1CUT/28<12 then
        do;
            FUP1CAT1="11-12 months";
            FUP1CA1N=12;
        end;
    end;
end;

```

```

end;

```

090177e196b7a9f0\FinalFinal On: 06-Apr-2021 21:45 (GMT)

;

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

ADD SAF1FL SAF2FL RAND1FL

DROP BOOSTER FLAG IF ALL NULL

```

*;
*****
;

data lbrna;

  set dataprot.lb;

  where lbtestcd="HIVR_US" and lbstat ne "NOT DONE" and index(visit,
    "V1_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

```

090177e196b7a9f0\\Final\\Final On: 06-Apr-2021 21:45 (GMT)

```

                                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;

                                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") 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 lbrna(keep=usubjid RNACAT RNACATN) lbcd4(keep=usubjid CD4CAT
CD4CATN);
by usubjid;
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;
```

```
data adsl;
```

```
set adsl;
label SAF1FL='Safety - exclud Multi-Enrolloer&HIV&IND'
      SAF2FL='Safety - exclude Multi-Enrolloer&IND'
      RAND1FL='Random - exclude Multi-Enrolloer';
```

```
if SAFFL="Y" and MULENRFL^="Y" and HIVFL^="Y" and trt01a ne "" then
  SAF1FL="Y";
else
  SAF1FL="N";

if SAFFL="Y" and MULENRFL^="Y" and trt01a ne "" then
  SAF2FL="Y";
else
  SAF2FL="N";

if RANDFL="Y" and MULENRFL^="Y" then
  RAND1FL="Y";
else
  RAND1FL="N";
```

```
run;
```

```
*****
```

```
Output datasets *;
```

\*\*\*\*\*  
;

Data datvprot.adsl(label="Subject-Level Analysis Dataset");

Retain StudyId UsubjId Subjid SiteId Age AgeU AAge AAgeU Sex SexN Race RaceN  
Ethnic EthnicN Raciald RacialdN Country SAFFL COMPLFL 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 ComplDt 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 vax202dt agetr01 agetru01 agetr02 agetru02 agetr03 agetru03  
agetgrln agetgrl AAgeY AAgeYu AAgeM AAgeMu AAgeW AAgeWu AAgeD AAgeDu AAgeH  
AAgeHu AgeGr1 AgeGr1N Arace AraceN RaceGr1 RaceGr1N Aethnic AethnicN EosDcDt  
EosDcRs EotDcDt EotDcRs;

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

090177e196b7a9f0\Final\Final On: 06-Apr-2021 21:45 (GMT)

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"  
 ComplFl Label="Completers Population Flag" ComplDt Label="Date of Completion"  
 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)" RacialDN

```

Label="Racial Designation (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)" EotDcDt
Label="End Of Treatment Discontinuation Date" Format=date9. EotDcRs
Label="End Of Treatment Discontinuation Reason" EosDcDt
Label="End Of Study Discontinuation Date" Format=date9. EosDcRs
Label="End Of Study Discontinuation Reason";

```

```
Drop _: Derived_ /*PC1MD2FL*/
```

```
ComplDt boostfl AAGE: AETHNIC: AGETGR1: AGETR02 AGETRU02
```

```

BE1DT: BE2DT: BE3DT: BE4DT: CDIFF: CODT: COMPLFL COVIS: DSRANGRP EFFICACY
ENRLDT ENRLNO IMMUNO INEX ISDT: MULTIPLE RACE1-RACE4 SAFETY SITEEXCLD
SRT1-SRT5 SRVLACDT TMPDTC TMPID VAX101DTM VAX102DTM VAX103DTM VAX201DTM
VAX202DTM VAX203DTM VIS: STEXCFL UNKRDFL Domain RfxStDtc RfxEnDtc RfxStDt
RfxEnDt RfxStTm RfxEnTm RfStDtc RfEnDtc RfIcDtc RfPEndtc BrthDtc;

```

```
Run;
```

```
proc printto;
```

```
run;
```

090177e196b7a9f0\Final\Final On: 06-Apr-2021 21:45 (GMT)