```
**********************************************************************************************
**;
**  Program Name    : adsympt.sas                                                             **;
**  Date Created    : 17Nov2021                                                               **;
**  Programmer Name : (b) (4), (b) (6)                                                        **;
**  Purpose         : Create adsympt dataset                                                  **;
**  Input data      : CE, CM, DS, FA, IS, HO, SUPPHO, MH, MB, SUPPMB, LB/LBOP, VS and ADSL    **;
**  Output data     : adsympt.sas7bdat                                                        **;
**********************************************************************************************
**;
%let
oprot=/Volumes/app/cdars/prod/sites/cdars4/prjC459/nda2_unblinded_esub/sbla1215_esub_sdtm/saseng/cdisc3_0/data/
sdtm;
%let
protori=/Volumes/app/cdars/prod/sites/cdars4/prjC459/nda2_unblinded_esub/sbla1215_esub_adam/saseng/cdisc3_0;
%let
prot=/Volumes/app/cdars/prod/sites/cdars4/prjC459/nda2_unblinded_esub/sbla1215_esub_adam/saseng/cdisc3_0/analy
sis/eSUB;

libname dataprot "&oprot." access=readonly;
libname datvprot "&protori./data_vai" access=readonly;
libname datvout "&prot./data_vai";
libname viewpx "/Volumes/app/saseng/prod/cdisc3_0/view/" access=readonly;

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

proc delete data=work._all_;
run;

** Get CE data. **;
data ce(keep=studyid domain usubjid paramn paramcd param parcat1 parcat2 aval
      avalc 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');
   length paramn 8 paramcd $8 param parcat1 parcat2 avalc $200;
   parcat1=upcase(strip(cecat));
   parcat2='';

   if upcase(strip(cescat))='SIGNIFICANT ACUTE RENAL DYSFUNCTION' then
      do;
         paramn=25;
         paramcd='SARDFN';
         param=upcase(strip(cescat));
      end;
   else if upcase(strip(cescat))='SIGNIFICANT ACUTE HEPATIC DYSFUNCTION' then
```

```
      do;
         paramn=30;
         paramcd='SAHDFN';
         param=upcase(strip(cescat));
      end;
   else if upcase(strip(cescat))='SIGNIFICANT ACUTE NEUROLOGIC DYSFUNCTION' then
      do;
         paramn=35;
         paramcd='SANDFN';
         param=upcase(strip(cescat));
      end;
   aval=.;
   avalc='Y';
   adt=input(cedtc, yymmdd10.);
   astdt=input(cestdtc, yymmdd10.);
   aendt=input(ceendtc, 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'))) and avalc ^='' then
         output;
run;

** Get CM data. **;

data cm;
   length cmscat $100.;
   set dataprot.cm;

   if cmscat='' then
      cmscat='';
run;

proc sort data=cm out=cm1;
   where upcase(strip(cmcat))='GENERAL CONCOMITANT MEDICATIONS' and
      upcase(strip(cmscat))='VASOPRESSORS AGENTS' and cmtrt ^='';
   by usubjid cmstdtc cmtrt;
run;

data cm(keep=studyid domain usubjid paramn paramcd param parcat1 parcat2 aval
      avalc adt astdt aendt visitnum visit);
   set cm1;
   by usubjid cmstdtc cmtrt;
   length paramn 8 paramcd $8 param parcat1 parcat2 avalc $200 visit $64;
   parcat1=strip(cmcat);
   parcat2=strip(cmscat);
   paramn=80;
   paramcd='VSOPRES';
   param=upcase(strip(cmscat));
   aval=.;

   if cmdecod ^='' then
      avalc=upcase(strip(cmdecod));
```

```
    else
       avalc=upcase(strip(cmtrt));
    adt=.;
    astdt=input(cmstdtc, yymmdd10.);
    aendt=input(cmendtc, yymmdd10.);
    visitnum=.;
    visit='';
    format adt astdt aendt date9.;

    if avalc ^='' then
       output;
    *if last.cmstdtc then output;
run;

** Gather death details DD. **;

data adsl_dth(keep=studyid usubjid dthdt);
    set datvprot.adsl;
    where dthdt ^=.;
run;

data dd(keep=studyid domain usubjid paramn paramcd param parcat1 parcat2 aval
        avalc adt astdt aendt visitnum visit);
    set adsl_dth(in=b);
    by usubjid;
    length paramn 8 paramcd $8 param parcat1 parcat2 avalc $200 visit $64;
    parcat2='';
    aval=.;
    adt=dthdt;
    astdt=.;
    aendt=.;
    visitnum=.;
    visit='';
    if b then
       do;
          domain='DD';
          paramn=95;
          paramcd='PRCDTH';
          param='PRIMARY CAUSE OF DEATH';
          parcat1='DEATH DETAILS CODED';
          avalc='';
       end;
    format adt astdt aendt date9.;
run;

** Get death records from DS. **;

proc sort data=dataprot.ds out=ds1;
    where upcase(strip(dsdecod))='DEATH';
    by usubjid dsdtc;
run;

data ds(keep=studyid domain usubjid paramn paramcd param parcat1 parcat2 aval
        avalc adt astdt aendt visitnum visit);
```

```
    set ds1;
    by usubjid dsdtc;

    if first.usubjid;
    length paramn 8 paramcd $8 param parcat1 parcat2 avalc $200 visit $64;
    paramn=99;
    paramcd=upcase(strip(dsdecod));
    param=upcase(strip(dsdecod));
    parcat1=upcase(strip(dscat));
    parcat2='';
    aval=.;
    avalc='Y';
    adt=.;
    astdt=input(dsstdtc, yymmdd10.);
    aendt=.;
    visitnum=.;
    visit='';
    format adt astdt aendt date9.;
run;

** Get FA data. **;

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

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

    if upcase(strip(fatestcd))='FSYMDATE' then
        output face_stdt;
    else if upcase(strip(fatestcd))='LSYMDATE' then
        output face_endt;
    else if upcase(strip(fatestcd))='SYMONGO' then
        output face_ong;
    else
        output face1;
run;

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

    if a;
run;

data fa(keep=studyid domain usubjid paramn paramcd param parcat1 parcat2 aval
        avalc adt astdt aendt visitnum visit) fa_excluded;
```

```
set face2;
length paramn 8 paramcd $8 param parcat1 parcat2 avalc $200;
param=upcase(strip(faobj));
parcat1='SIGNS AND SYMPTOMS OF DISEASE';
parcat2='RESPIRATORY ILLNESS';
avalc=strip(fastresc);

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

      if paramcd='CHILLS' then
         paramn=1;

      if paramcd='DIARRHEA' then
         paramn=2;

      if paramcd='FEVER' then
         paramn=3;
   end;
else if strip(param)='NEW LOSS OF TASTE OR SMELL' then
   do;
      paramn=4;
      paramcd='NLTSTSML';
   end;
else if strip(param)='NEW OR INCREASED COUGH' then
   do;
      paramn=5;
      paramcd='NCOUG';
   end;
else if strip(param)='NEW OR INCREASED MUSCLE PAIN' then
   do;
      paramn=6;
      paramcd='NMUSPN';
   end;
else if strip(param)='NEW OR INCREASED SHORTNESS OF BREATH' then
   do;
      paramn=7;
      paramcd='NSTBRTH';
   end;
else if strip(param)='NEW OR INCREASED SORE THROAT' then
   do;
      paramn=8;
      paramcd='NSRTHROT';
   end;
else if strip(param)='VOMITING' then
   do;
      paramn=9;
      paramcd='VOMIT';
   end;
else if strip(param)='LOSS OF TASTE/SMELL' then
   do;
      paramn=10;
      paramcd='LSTSTSML';
```

```
   end;
else if strip(param) in ('NEW OR INCREASED NASAL CONGESTION',
   'NASAL CONGESTION') then
      do;
      paramn=11;
      paramcd='NNSLCONG';
      param='NEW OR INCREASED NASAL CONGESTION';
   end;
else if strip(param)='NEW OR INCREASED NASAL DISCHARGE' then
   do;
      paramn=12;
      paramcd='NNSLDSCH';
   end;
else if strip(param)='NEW OR INCREASED SPUTUM PRODUCTION' then
   do;
      paramn=13;
      paramcd='SPUTPROD';
   end;
else if strip(param) in ('NEW OR INCREASED WHEEZING', 'WHEEZING') then
   do;
      paramn=14;
      paramcd='WHEEZ';
      param='NEW OR INCREASED WHEEZING';
   end;
else if strip(param)='FATIGUE' then
   do;
      paramn=15;
      paramcd='FATIGUE';
      param='FATIGUE';
   end;
else if strip(param)='HEADACHE' then
   do;
      paramn=16;
      paramcd='HEADACHE';
      param='HEADACHE';
   end;
else if strip(param)='NAUSEA' then
   do;
      paramn=18;
      paramcd='NAUSEA';
      param='NAUSEA';
   end;
else
   do;
      id=prxparse('/' || 'RUNNY NOSE' || '/i');
      call prxsubstr(id, param, point, lng);

      if lng > 0 or upcase(faobj)='RHINORRHOEA' then
         do;
            paramn=17;
            paramcd='RIHNRA';
            param='RHINORRHOEA';
         end;
   end;
```

```
    aval=.;
    adt=input(fadtc, yymmdd10.);
    astdt=input(fastdtc, ??yymmdd10.);
    aendt=input(faendtc, ??yymmdd10.);
    format adt astdt aendt date9.;

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

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

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

** Get data from HO. **;

proc sql;
    create table ho1 as select * from dataprot.ho left join
             (select qval as HCUHSP 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 qval as HCUICU from dataprot.suppho as b where
       upcase(strip(qnam))='HCUICU') on strip(usubjid)=strip(b.usubjid) and
       strip(put(hoseq, best.))=strip(b.idvarval);
    create table ho3 as select distinct * from ho2 left join
             (select hostdtc as hostdtc_, hoendtc as hoendtc_, hoenrtpt as
       hoenrtpt_, hoentpt as hoentpt_ from ho2 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
                (hcuhsp ^='' or hcuicu ^='') order by usubjid, hoseq, hostdtc;
quit;

data ho4;
    set ho3;
    by usubjid hoseq hostdtc;

    if first.hoseq and last.hoseq then
       dupflg=0;
    else
       dupflg=1;
run;

data ho5;
```

```
    set ho4;
    by usubjid hoseq hostdtc;

    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;

data ho(keep=studyid domain usubjid paramn paramcd param parcat1 parcat2 aval
       avalc adt astdt aendt visitnum visit);
    set ho5;
    by usubjid hoseq hostdtc;
    where upcase(strip(hoterm))='ICU' or hcuhsp ^='' or hcuicu ^='';
    length paramn 8 paramcd $8 param parcat1 parcat2 avalc $200;
    parcat1=strip(hocat);
    parcat2='';
    aval=.;
    adt=input(hodtc, yymmdd10.);
    astdt=input(hostdtc, yymmdd10.);
    aendt=input(hoendtc, yymmdd10.);
    format adt astdt aendt date9.;

    if upcase(strip(hoterm))='ICU' then
       do;
          paramn=91;
          paramcd='HCUICU';
          param='SUBJECT IN ICU DUE TO POTENTIAL COVID-19 ILLNESS';
          avalc='Y';
          output;
       end;

    if upcase(strip(hcuicu))='Y' then
       do;
          paramn=91;
          paramcd='HCUICU';
          param='SUBJECT IN ICU DUE TO POTENTIAL COVID-19 ILLNESS';
          avalc=upcase(strip(hcuicu));
          astdt=input(hostdtc_, yymmdd10.);
          aendt=input(hoendtc_, yymmdd10.);
          output;
       end;

    if upcase(strip(hcuhsp))='Y' then
       do;
          paramn=92;
          paramcd='HCUHSP';
          param='HOSPITALIZED DUE TO COVID-19 ILLNESS?';
          avalc=upcase(strip(hcuhsp));
          astdt=input(hostdtc_, yymmdd10.);
          aendt=input(hoendtc_, yymmdd10.);
          output;
       end;
run;
```

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

data is(keep=studyid domain usubjid paramn paramcd param parcat1 parcat2 aval
      avalc adt astdt aendt visitnum visit isspec ismethod);
   set dataprot.is;
   where strip(istestcd) in ('C19NIG');
   length paramn 8 paramcd $8 param parcat1 parcat2 avalc $200;
   parcat1=strip(iscat);
   parcat2='';
   paramn=90;
   paramcd=strip(istestcd);
   param=upcase(strip(istest));
   aval=.;
   avalc=upcase(strip(isorres));
   adt=input(isdtc, yymmdd10.);
   astdt=.;
   aendt=.;
   format adt astdt aendt date9.;
   *if strip(visit) in ('V1_DAY1_VAX1_L') then output;
run;

data lb(keep=studyid domain usubjid paramn paramcd param parcat1 parcat2 aval
      avalc adt astdt aendt visitnum visit_ lborresu rename=(visit_=visit));
   set dataprot.lb;
   where strip(lbtestcd)='PO2FIO2';
   length paramn 8 paramcd $8 param parcat1 parcat2 avalc $200 visit_ $64;
   parcat1=strip(lbcat);
   parcat2=strip(lbscat);
   paramn=60;
   paramcd=strip(lbtestcd);
   param=upcase(strip(lbtest));
   aval=input(lborres, best.);
   avalc='';
   visit_=strip(visit);
   adt=input(lbdtc, 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'))) and aval ^=. then
         output;
run;

** Get MB data. **;

data mb(keep=studyid domain usubjid paramn paramcd param parcat1 parcat2 aval
      avalc adt astdt aendt visitnum visit_ mbloc mbspec mbmethod
      rename=(visit_=visit));
   set dataprot.mb;
   where (upcase(strip(mbtest))='SEVERE ACUTE RESP SYNDROME CORONAVIRUS 2' and
      upcase(strip(mbmethod))="IMMUNOCHROMATOGRAPHY") or
```

```
        (upcase(strip(mbtest)) in ('CEPHEID RT-PCR ASSAY FOR SARS-COV-2',
      'CEPHEID RT-PCR ASSAY OF SARS-COV-2') and
      upcase(strip(mbmethod))='REVERSE TRANSCRIPTASE PCR');
   length paramn 8 paramcd $8 param parcat1 parcat2 avalc $200 visit_ $64;

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

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

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

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

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

data mh;
   set dataprot.mh;

   if strip(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');
   c19ilhfl='Y';
run;

** Get VS data. **;

data vs(keep=studyid domain usubjid paramn paramcd param parcat1 aval avalc adt
      astdt aendt visitnum visit vsstresu);
```

```
    set dataprot.vs;
    where upcase(strip(vscat))='GENERAL VITAL SIGNS' and strip(vstestcd)
       in ('RESP', 'HR', 'OXYSAT', 'DIABP', 'SYSBP');
    length paramn 8 paramcd $8 param parcat1 avalc $200;
    paramcd=strip(vstestcd);
    param=upcase(strip(vstest));
    parcat1=strip(vscat);
    parcat2=strip(vsscat);

    if vstestcd='RESP' then
       paramn=50;

    if vstestcd='HR' then
       paramn=51;

    if vstestcd='OXYSAT' then
       paramn=52;

    if vstestcd='DIABP' then
       paramn=53;

    if vstestcd='SYSBP' then
       paramn=54;
    aval=vsstresn;

    if aval=. and vsstresc ^='' then
       avalc=strip(vsstresc);
    adt=input(vsdtc, 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'))) and (aval ^=. or avalc ^='')
       then
          output;
run;

data adsympt1;
    set ce cm dd ds fa ho is lb mb1 vs;
    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;

** To collect visits from Signs and Symptoms, Covid test results, medication/procedures, ICU admission, vital signs
and
** create an aggregated set of visits and collapse when they are within specified range;
%let __excl_vis1a =
```

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

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

** Get FA data. **;

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

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

   if upcase(strip(fatestcd))='FSYMDATE' then
      output __face_stdt;
   else if upcase(strip(fatestcd))='LSYMDATE' then
      output __face_endt;
   else if upcase(strip(fatestcd))='SYMONGO' then
      output __face_ong;
   else
      output __face1;
run;

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

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

** Get data from HO. **;

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

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

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

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

** Get IS data. **;

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

```
** Get LB data. **;

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

** Get MB data. **;

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

** Get VS data. **;

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

data __visits_sdtm;
   set __ce __fa __is __lb __mb __vs;
run;

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

data __visits_sdtm_rv1(drop=visit_) __visits_sdtm_rv1a(drop=visit_ visitnum
      covid_vis_cnt) __covid_vis_cnt(keep=usubjid covid_vis_cnt);
```

```
    set __visits_sdtm;
    by usubjid visitnum visit adt astdt aendt domain;
    visitnum_bak=visitnum;
    visit_bak=strip(visit);

    if length(visit) >=8 then
       do;

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

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

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

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

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

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

proc sql;
    ** Get visitnums for repeat visits. **;
    create table __visits_sdtm_rv2a as select distinct * from
             (select * from __visits_sdtm_rv1a) left join
             (select visitnum from __visits_sdtm_rv1 as b where rvflg ^=1) on
       usubjid=b.usubjid and visit=b.visit;
    ** Check if any of them missing visitnum from above. **;
    create table __visits_sdtm_rv3a as select * from __visits_sdtm_rv2a left join
             (select visitnum as visitnum_rv, visit as visit_rv, astdt as astdt_rv,
       aendt as aendt_rv from __visits_sdtm_rv1 as b where domain='FA' and
       astdt ^=. and aendt ^=.) on usubjid=b.usubjid and b.astdt <=adt <=b.aendt and
       visitnum=.;
    ** Get visits count to assign visitnums. **;
```

```
    create table __visits_sdtm_rv4a as select * from __visits_sdtm_rv3a left
        join (select covid_vis_cnt from __covid_vis_cnt as b) on usubjid=b.usubjid
        order by domain, usubjid, visitnum, visit, adt, astdt, aendt;
quit;

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

    if visitnum=. then
        do;

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

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

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

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

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

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

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

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

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

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

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

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

   if a and b then
      eligflg=1;
run;

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

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

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

data __visits3(drop=mlvisflg usubjid_same_dt) __visits3a(drop=mlvisflg
      usubjid_same_dt clsp_pros_flg);
   set __visits2a;
   by usubjid astdt descending aendt visitnum;
   where mlvisflg=1 or usubjid_same_dt ^='';

   if (domain='FA') or (domain='HO' and astdt ^=. and aendt ^=.) or (domain='VS'
      and astdt ^=.) then
         do;
```

```
            clsp_pros_flg=1;
            output __visits3a;
        end;
    output __visits3;
run;

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

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

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

    if usubjid=usubjid_nxt then
        do;

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

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

                    if aendt_ < astdt_nxt then
                        aendt_=astdt_nxt;

                    if aendt_nxt ^=. and aendt_ < aendt_nxt then
                        aendt_=aendt_nxt;
                end;
            ** Check the current dates and visits and collapse. **;
            ** AD(26May2021): added '. <' condition to not collapse visit is starting date in missing for next visit **;

            if visitnum_ ^=visitnum then
                do;

                    if (aendt_=. and astdt_ <=astdt <=sum(astdt_, 3))
```

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

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

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

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

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

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

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

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

   if avisitn=. then
      do;
         avisitn=visitnum;
         avisit=visit;
      end;
```

```
    if last.recseq then
       keepflg=1;
run;

** Prepare all visits. **;

data __visits_raw_prepare;
    set __visits_raw;

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

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

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

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

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

    if not(visflg=1 and mlvisflg=1) then
       do;
          avisitn=visitnum;
          avisit=strip(visit);
       end;

    if rvflg=1 then
```

```
        do;
           visitnum=visitnum_bak;
           visit=visit_bak;

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

     if adtflg=1 then
        astdt=.;

     if rvflg=1 then
        clspfl='Y';

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

  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 order by usubjid, visitnum, visit, adt,
        astdt, aendt;
  quit;

  data adsympt4;
     recseq=put(_n_, z7.);
     set adsympt3;

     if clspfl='Y' then
        do;
           avisitn=avisitn_clsp;
           avisit=avisit_clsp;
        end;
  run;

  ** Generate unique visits. **;

  proc sql;
     create table visits_mod as select usubjid, avisitn, avisit, min(astdt) as
        astdt_v format yymmdd10., max(aendt) as aendt_v format yymmdd10. from
        adsympt4 where astdt ^=. and avisitn ^=. and avisit ^='' group by usubjid,
        avisitn, avisit;
  quit;

  data visits_mod;
     set visits_mod;
     by usubjid avisitn avisit;

     if first.usubjid and last.usubjid then
        viscntflg=0;
```

```
   else
      viscntflg=1;
run;

proc sql;
   ** Merge Covid-19 illness history flag. **;
   create table adsympt5(drop=avisitn_clsp avisit_clsp clspfl) as select * from
      adsympt4 left join
            (select distinct usubjid as usub, c19ilhfl from mh as b where
      c19ilhfl ^='') on strip(usubjid)=strip(b.usubjid);
   ** Try to map AVISTN and AVISIT for records with missing visits. **;
   create table adsympt5_b as select * from
            (select * from adsympt4 where visitnum=.) left join
            (select viscntflg, avisitn as avisitn_v, avisit as avisit_v, astdt_v,
      aendt_v from visits_mod as b) on
               (usubjid=b.usubjid) and
               (((astdt ^=.) and ((. < astdt_v <=astdt) or (aendt ^=. and
      aendt_v ^=. and astdt_v <=astdt <=aendt_v) or
                (aendt ^=. and b.viscntflg=0 and astdt_v <=astdt))) or
               ((astdt ^=. and adt ^=.) and ((aendt=. and astdt_v <=adt)
      or (aendt ^=. and astdt_v <=adt <=aendt_v))) ) order by usubjid, astdt, adt,
      recseq, astdt_v;
quit;

data adsympt5_c;
   set adsympt5_b;
   by usubjid astdt adt recseq astdt_v;

   if strip(paramcd) not in ('PRCDTH', 'SECDTH', 'DEATH') then
      do;
         avisitn=avisitn_v;
         avisit=avisit_v;
      end;

   if first.recseq and last.recseq then
      keepflg=1;
   else
      do;

         if last.recseq then
            keepflg=1;
      end;
run;

data adsympt6(drop=recseq avisitn_v avisit_v astdt_v aendt_v keepflg);
   set adsympt5(where=(visitnum ^=.)) adsympt5_c(where=(keepflg=1));
run;

proc sql;
   ** Merge ADSL Variables. **;
   create table adsympt7 as select * from adsympt6 left join
            (select siteid, subjid, brthdt, agegr1n, agegr1, sex, race, ethnic,
      armcd, arm, actarmcd, actarm, randdt, trtsdt, trtedt, vax101dt, vax102dt,
      enrlfl, randfl, saffl, evaleffl, ev14effl, aai1effl, aai2effl, phasen from
```

```
      datvprot.adsl as b) on strip(usubjid)=strip(b.usubjid) order by studyid,
      usubjid, visitnum, adt, astdt, paramn;
quit;

proc sql;
   ** Keep only 12-15 years of Age subjects **;
   create table adsympt71 as select * from adsympt7 where usubjid in (select
      distinct usubjid from datvprot.adsl where 12 <=agetr01 <=15) order by
      studyid, usubjid, visitnum, adt, astdt, paramn;
quit;

data adsympt8;
   set adsympt71;
   by studyid usubjid visitnum adt astdt paramn;
   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';
   call missing(ady, astdy, aendy);

   if trtsdt ^=. then
      do;

         if adt ^=. then
            do;
               ady=adt - trtsdt;

               if adt >=trtsdt then
                  ady=ady + 1;
            end;

         if astdt ^=. then
            do;
               astdy=astdt - trtsdt;

               if astdt >=trtsdt then
                  astdy=astdy + 1;
            end;

         if aendt ^=. then
            do;
               aendy=aendt - trtsdt;
```

```
            if aendt >=trtsdt then
               aendy=aendy + 1;
         end;
      end;

   if siteid='' then
      siteid=strip(scan(usubjid, 2, ' '));

   if subjid='' then
      subjid=strip(scan(usubjid, 3, ' '));
run;

data datvout.adsympt(label="Covid-19 Signs and Symptoms Analysis Dataset");
   retain studyid usubjid siteid subjid brthdt agegr1n agegr1 sex race ethnic
      armcd arm actarmcd actarm paramn paramcd param parcat1 parcat2 aval avalc
      visitnum visit avisitn avisit adt ady astdt astdy aendt aendy isspec ismethod
      mbloc mbspec mbmethod vsstresu randdt trtsdt trtedt vax101dt vax102dt enrlfl
      randfl saffl evaleffl ev14effl aai1effl aai2effl c19ilhfl;
   set adsympt8;
   where phasen > 1;
   by studyid usubjid visitnum adt astdt paramn;
   keep studyid usubjid siteid subjid brthdt agegr1n agegr1 sex race ethnic armcd
      arm actarmcd actarm paramn paramcd param parcat1 parcat2 aval avalc visitnum
      visit avisitn avisit adt ady astdt astdy aendt aendy isspec ismethod mbloc
      mbspec mbmethod vsstresu randdt trtsdt trtedt vax101dt vax102dt enrlfl randfl
      saffl evaleffl ev14effl aai1effl aai2effl c19ilhfl;
   label paramn='Parameter (N)' paramcd='Parameter Code' param='Parameter'
      parcat1='Parameter Category 1' parcat2='Parameter Category 2'
      aval='Analysis Value' avalc='Analysis Value (C)' avisitn='Analysis Visit (N)'
      avisit='Analysis Visit' adt='Analysis Date' ady='Analysis Relative Day'
      astdt='Analysis Start Date' astdy='Analysis Start Relative Day'
      aendt='Analysis End Date' aendy='Analysis End Relative Day'
      isspec='IS Specimen Type' ismethod='IS Method of Test or Examination'
      mbspec='MB Specimen Type' mbmethod='MB Method of Test or Examination'
      c19ilhfl='Prior Covid-19 Illness History Flag';
run;

proc printto;
run;
```