

```

*****
*****;
** Program Name   : adva.sas
** Date Created   : 07Mar2021
** Programmer Name: (b) (6)
** Purpose        : Create adva dataset
** Input data     : is suppis adsl
** Output data    : adva.sas7bdat
*****
*****;

%let
oprot=/Volumes/app/cdars/prod/sites/cdars4/prjC459/nda2_unblinded_esub/bla_euaext_esub_sdtm/saseng/cdisc3_0;
%let prot=/Volumes/app/cdars/prod/sites/cdars4/prjC459/nda2_unblinded_esub/euaext_esub_adam/saseng/cdisc3_0;
libname dataprot "&oprot./data" access=readonly;
libname datvprot "&prot./data_vai";

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

*****;
* Clean *;
*****;

proc delete data=work._all_;
run;

*****;
* Read in source IS SDTM datasets. *;
*****;

/*Impute time to correct period */
data is;
  length isdte $20;
  set dataprot.is;
  where not missing(istested);

  if (index(visit, "COVID") or visit in ('V101_VAX3' 'V201_SURVEIL_CONSENT'))
    then
      isdte=strip(isdte)||"T23:59:59";
  else
      isdte=strip(isdte)||"T00:00:00";
run;

/*Select retest result for Phase 1 subjects with available retests*/
data is_;
  set is;

  if ISTSTDTL ne " ";
  keep usubjid istest istested visit visitnum isdte ISTSTDTL;
run;

data is_;

```

```

set is_;

proc sort nodupkey;
  by usubjid visit;
run;

proc sql;
  create table is2_ as select a.*, b.ISTSTDTL as _ISTSTDTL1 from is as a left
    join is_ as b on a.usubjid=b.usubjid and a.visit=b.visit and
      a.visitnum=b.visitnum;
quit;

data is;
  set is2_;
  ISUSE="N";

  /*have retest, retest value not missing*/
  if _ISTSTDTL1 ne " " and ISTSTDTL ne " " then
    ISUSE="Y";

  if _ISTSTDTL1=" " and ISTSTDTL=" " then
    ISUSE="Y";
run;

data _dsnin;
  set is;
run;

data _dsnin;
  set _dsnin;
  ADT=input(ISDTC, ??is8601da.);

  if ^missing(ISDTC) then
    do;
      length hr mn sc $2 newtime $8;
      yr=substr(ISDTC, 1, 4);
      hr=substr(ISDTC, 12, 2);
      mn=substr(ISDTC, 15, 2);
      sc=substr(ISDTC, 18, 2);
      ;

      if yr ne ' ' then
        do;

          if hr eq " " then
            do;
              newtime="00:00:00";
              ATM=input(newtime, ??time8.);
              format ATM time8.;
              ATMF='H';
            end;
          else
            do;
              tflag=' ';

```

```

        if sc eq " " then
            do;
                sc='00';
                tflag='S';
            end;

        if mn eq " " then
            do;
                mn='00';
                tflag='M';
            end;
        newtime=(trim(left(hr))||':'||trim(left(mn))||':'||trim(left(sc)));
        ATM=input(newtime, ??time8.);
        format ATM time8.;
        ATMF=tflag;
        drop tflag;
    end;
end;
drop yr hr mn sc newtime;
end;
;
ADTM=DHMS(ADT, 0, 0, ATM);
format ADT date9. ADTM datetime20. ATM time8.;
label ADT="Analysis Date" ADTM="Analysis Date/Time" ATM="Analysis Time"
       ATMF="Analysis Time Imputation Flag";
run;

/* Remove NOTDONEs and create PARAM */
data _dsnin;
    set _dsnin(rename=(isstresn=aval iscat=parcat1 isdy=ady isseq=srcseq
        domain=srdom istested=paramcd)) end=eof;
    where upcase(paramcd) ne "ISALL";

    if paramcd in ("C2NGNT50" "C2NGNT90") then
        do;

            if not missing(ismethod) then
                param=trim(left(istest))||' ('||trim(left("titer"))||') - '||trim(left(ismethod));
            else
                param=trim(left(istest))||' ('||trim(left("titer"))||')';
            end;
        else if paramcd in ("C19NIG") then
            do;

                if not missing(ismethod) then
                    param=trim(left(istest))||' - '||trim(left(ismethod));
                else
                    param=trim(left(istest));
                end;
            else
                do;

                    if not missing(ismethod) then

```

```

        param=trim(left(istest))||' ('||trim(left(isstresu))||') - '||trim(left(ismethod));
    else
        param=trim(left(istest))||' ('||trim(left(isstresu))||')';
    end;
    avalc=isstresc;

    /* if missing(isstresu) and isorres in ("IND" "QNS" "INDETERMINATE" "NOT DONE") then delete; */ /*Check
before finalization*/

    if isorres in ("NOT DONE") then
        delete;
    srcvar="ISSTRESN";

    if parcat1="SEROLOGY" then
        parcat1n=1;
run;

/* Merge with ADSL */
Data work.adsl;
Set datvprot.adsl;

If Missing(TR01STM) then
    _apx_TR01STM="23:59:30"t;
Else
    _apx_TR01STM=TR01STM;

If Missing(TR01ETM) then
    _apx_TR01ETM="23:59:29"t;
Else
    _apx_TR01ETM=TR01ETM;

If ^Missing(TR01SDT) then
    _apx_TR01SDTM=dhms(TR01SDT, 0, 0, _apx_TR01STM);

If ^Missing(TR01EDT) then
    _apx_TR01EDTM=dhms(TR01EDT, 0, 0, _apx_TR01ETM);
AP01SDT=datepart(_apx_TR01SDTM);
AP01STM=timepart(_apx_TR01SDTM);

If Missing(TR02STM) then
    _apx_TR02STM="23:59:30"t;
Else
    _apx_TR02STM=TR02STM;

If Missing(TR02ETM) then
    _apx_TR02ETM="23:59:29"t;
Else
    _apx_TR02ETM=TR02ETM;

If ^Missing(TR02SDT) then
    _apx_TR02SDTM=dhms(TR02SDT, 0, 0, _apx_TR02STM);

If ^Missing(TR02EDT) then
    _apx_TR02EDTM=dhms(TR02EDT, 0, 0, _apx_TR02ETM);

```

```

AP02SDT=datepart(_apx_TR02SDTM);
AP02STM=timepart(_apx_TR02SDTM);

if ^Missing(_apx_TR02SDTM-1) then
  do;
    _apx_TR01EDTM=min((_apx_TR02SDTM-1), (_apx_TR01EDTM+((365)*86400)));
  end;
else
  _apx_TR01EDTM=_apx_TR01EDTM+((365)*86400);
AP01EDT=datepart(_apx_TR01EDTM);
AP01ETM=Timepart(_apx_TR01EDTM);
AP01SDTM=dhms(AP01SDT, 0, 0, AP01STM);
AP01EDTM=dhms(AP01EDT, 0, 0, AP01ETM);
Attrib AP01SDT Label="Period 01 Start Date" AP01EDT Label="Period 01 End Date"
  AP01STM Label="Period 01 Start Time" AP01ETM Label="Period 01 End Time"
  AP01SDTM Label="Period 01 Start Datetime" AP01EDTM
  Label="Period 01 End Datetime";
Format AP01SDT AP01EDT date9. AP01STM AP01ETM time8. AP01SDTM AP01EDTM
  datetime20.;
_apx_TR02EDTM=_apx_TR02EDTM+((365)*86400);
AP02EDT=datepart(_apx_TR02EDTM);
AP02ETM=Timepart(_apx_TR02EDTM);
AP02SDTM=dhms(AP02SDT, 0, 0, AP02STM);
AP02EDTM=dhms(AP02EDT, 0, 0, AP02ETM);
Attrib AP02SDT Label="Period 02 Start Date" AP02EDT Label="Period 02 End Date"
  AP02STM Label="Period 02 Start Time" AP02ETM Label="Period 02 End Time"
  AP02SDTM Label="Period 02 Start Datetime" AP02EDTM
  Label="Period 02 End Datetime";
Format AP02SDT AP02EDT date9. AP02STM AP02ETM time8. AP02SDTM AP02EDTM
  datetime20.;

run;

proc sort data=_dsnin out=_ds1;
  by USUBJID;
run;

proc sort data=adsl out=_ds2;
  by USUBJID;
run;

data adsl_dsin;
  merge _ds1(in=d1) _ds2(in=d2 keep=Usubjid SUBJID SITEID AGE AGEU SEX SEXN RACE
    RACEN RANDFL SAFFL ARM ARMCD ACTARM ACTARMCD TRTSDT TRTEDT TRTSTM TRTETM
    TRT01A TRT01AN TRT02A TRT02AN TRT01P TRT01PN TRT02P TRT02PN TR01SDT TR01STM
    TR01SDTM TR01EDT TR01ETM TR01EDTM TR02SDT TR02STM TR02SDTM TR02EDT TR02ETM
    TR02EDTM COHORTN COHORT VAX101DT VAX102DT VAX10UDT VAX201DT VAX202DT SAFFL
    AAI01FL EVAL02FL AAI02FL EVAL01FL DOSALVL DOSALVLN DOSPLVL DOSPLVLN AGEGR1
    AGEGR1N AGEGR2 AGEGR2N AGEGR4 AGEGR4N PHASE PHASEN COVBLST HIVFL PEDIMMFL
    EV1MD2FL AGETR01 VAX101 VAX102 VAX10U VAX201 VAX202 VAX20U VAX20UDT UNBLNDDT
    MULENRFL RAND1FL TRTSDTM TRTEDTM AP01SDT AP01STM AP01SDTM AP01EDT AP01ETM
    AP01EDTM AP02SDT AP02STM AP02SDTM AP02EDT AP02ETM AP02EDTM TR01SDT TR01STM
    TR01SDTM TR01EDT TR01ETM TR01EDTM TR02SDT TR02STM TR02SDTM TR02EDT TR02ETM
    TR02EDTM);
  by USUBJID;

```

```

    if d1;
run;

/* Create period related variables */
data adsl_dsin;
    set adsl_dsin;
    _AP01SDT=AP01SDT;
    _AP01STM=AP01STM;
    _AP01SDTM=AP01SDTM;
    _AP01EDT=AP01EDT;
    _AP01ETM=AP01ETM;
    _AP01EDTM=AP01EDTM;
    AP01SDTM=AP01SDTM;
    AP01EDTM=AP01EDTM - hms(0, 0, 0);
    AP01SDT=datepart(AP01SDTM);
    AP01STM=timepart(AP01SDTM);
    AP01EDT=datepart(AP01EDTM);
    AP01ETM=timepart(AP01EDTM);
    _AP02SDT=AP02SDT;
    _AP02STM=AP02STM;
    _AP02SDTM=AP02SDTM;
    _AP02EDT=AP02EDT;
    _AP02ETM=AP02ETM;
    _AP02EDTM=AP02EDTM;
    AP02SDTM=AP02SDTM - hms(0, 0, 0);
    AP02EDTM=AP02EDTM;
    AP02SDT=datepart(AP02SDTM);
    AP02STM=timepart(AP02SDTM);
    AP02EDT=datepart(AP02EDTM);
    AP02ETM=timepart(AP02EDTM);
run;

data _dsnin;
    attrib APERIODC length=$20. label="Period (C)" APERIOD label="Period" APHASE
        length=$20. label="Phase" TRTP length=$100. label="Planned Treatment" TRTA
        length=$100. label="Actual Treatment" TRTPN label="Planned Treatment (N)"
        TRTAN label="Actual Treatment (N)" APERSTM label="Period Start Time"
        format=time8. APERETM label="Period End Time" format=time8. APERSDTM
        label="Period Start Datetime" format=datetime20. APEREDTM
        label="Period End Datetime" format=datetime20. APERSDT
        label="Period Start Date" format=date9. APEREDT label="Period End Date"
        format=date9. APERDY label="Relative Day of Period Start";
    set adsl_dsin;
    Array APSDT[*] AP01SDT AP02SDT;
    Array APEDT[*] AP01EDT AP02EDT;
    Array APSDTM[*] AP01SDTM AP02SDTM;
    Array APEDTM[*] AP01EDTM AP02EDTM;

    if ADTM <=APEDTM[1] and ^missing(ADTM) and ^missing(APEDTM[1])then
        do;
            TRTP=TRT01P;
            TRTA=TRT01A;
            TRTPN=TRT01PN;

```

```

    TRTAN=TRT01AN;
    APERIOD=1;
    APERIODC="Period 01";
end;
else if ADTM > APEDTM[1] and ^missing(ADTM) and ^missing(APEDTM[1]) then
do;
    TRTP=TRT02P;
    TRTA=TRT02A;
    TRTPN=TRT02PN;
    TRTAN=TRT02AN;
    APERIOD=2;
    APERIODC="Period 02";
end;

if ADTM <=APSDTM[1] and ^missing(ADTM) and ^missing(APSDTM[1]) then
do;
    Aphase="PRE-TREATMENT";
end;
else if APSDTM[1] < ADTM <=APEDTM[1] and ^missing(ADTM)
and ^missing(APSDTM[1]) and ^missing(APEDTM[1]) then
do;
    Aphase="TREATMENT "||"01";
end;
else if APEDTM[1] < ADTM <=APSDTM[2] and ^missing(APEDTM[1])
and ^missing(ADTM) and ^missing(APSDTM[2]) then
do;
    Aphase="OFFDRUG "||"01";
end;
else if APSDTM[2] < ADTM <=APEDTM[2] and ^missing(ADTM)
and ^missing(APSDTM[2]) and ^missing(APEDTM[2]) then
do;
    Aphase="TREATMENT "||"02";
end;
else if ADTM > APEDTM[2] and ^missing(ADTM) and ^missing(APEDTM[2]) then
do;
    Aphase="FOLLOW-UP";
end;

if ADTM <=APEDTM[1] and ^missing(ADTM) and ^missing(APEDTM[1]) then
do;

    if ADT < APSDT[1] then
        APERDY=ADT - APSDT[1];
    else if ADT >=APSDT[1] then
        APERDY=ADT - APSDT[1] + 1;
    end;
else if ADTM > APEDTM[1] and ^missing(ADTM) and ^missing(APEDTM[1]) then
do;

    if ADT < APSDT[2] and ^missing(APSDT[2]) then
        APERDY=ADT - APSDT[2];
    else if ADT >=APSDT[2] and ^missing(APSDT[2]) then
        APERDY=ADT - APSDT[2] + 1;
    end;
end;

```

```

if ADTM <= APEDTM[1] and ^missing(ADTM) and ^missing(APEDTM[1]) then
do;
  APERSDT=AP01SDT;
  APEREDT=AP01EDT;
  APERSTM=AP01STM;
  APERETM=AP01ETM;
  APERSDTM=AP01SDTM;
  APEREDTM=AP01EDTM;
end;
else if ADTM > APEDTM[1] and ^missing(ADTM) and ^missing(APEDTM[1]) then
do;
  APERSDT=AP02SDT;
  APEREDT=AP02EDT;
  APERSTM=AP02STM;
  APERETM=AP02ETM;
  APERSDTM=AP02SDTM;
  APEREDTM=AP02EDTM;
end;
run;

run;

data adsl_dsin;
  attrib AP01SDT format=date9. label="Period 01 Start Date" AP01EDT
    format=date9. label="Period 01 End Date" AP01STM
    format=time8. label="Period 01 Start Time" AP01ETM
    format=time8. label="Period 01 End Time" AP01SDTM
    format=datetime20. label="Period 01 Start Datetime" AP01EDTM
    format=datetime20. label="Period 01 End Date/Time" AP02SDT
    format=date9. label="Period 02 Start Date" AP02EDT
    format=date9. label="Period 02 End Date" AP02STM
    format=time8. label="Period 02 Start Time" AP02ETM
    format=time8. label="Period 02 End Time" AP02SDTM
    format=datetime20. label="Period 02 Start Datetime" AP02EDTM
    format=datetime20. label="Period 02 End Date/Time";
  set _dsnin(drop=AP01SDT AP01STM AP01SDTM AP01EDT AP01ETM AP01EDTM AP02SDT
    AP02STM AP02SDTM AP02EDT AP02ETM AP02EDTM rename=( _AP01SDT=AP01SDT
    _AP01STM=AP01STM _AP01SDTM=AP01SDTM _AP01EDT=AP01EDT _AP01ETM=AP01ETM
    _AP01EDTM=AP01EDTM _AP02SDT=AP02SDT _AP02STM=AP02STM _AP02SDTM=AP02SDTM
    _AP02EDT=AP02EDT _AP02ETM=AP02ETM _AP02EDTM=AP02EDTM) );
run;

data Allparamdata;
  length XOCFDERV 8;
  set adsl_dsin;
  paramcd=upcase(paramcd);
  winderv=0;
  XOCFDERV=0;
  label XOCFDERV="Result imputed by XOCF method (1=Yes)";
run;

data ParQData1;
  Length WINTNAME WINTMPLT WINFNAME $40 WINMETH $10 XVISWINM $30 XOCFMETH $20

```



```

XOCFSUB XOCFWIN $400 VISWINM obs 8;
set Allparamdata;
WinTmplt="VISITS";
WinTName="VISITS";
WinMeth="L";
XVisWinM="";
VisWinM=0;
XOCFMeth="OBS";
XOCFSub="";
XOCFWin="";
WinFName="winmetal";
obs=_n_;
Label WinTName="Window template name" WinMeth="Windowing selection method"
      XOCFMeth="XOCF selection method" Wintmplt="Original Window Template"
      XOCFSub="Criteria to select obs for Carry Forward"
      XOCFWin="Criteria to select window for Imputation";
run;

```

```

/* visit window */
data ParQCond1;
  length avisit $ 40 baseinfo $ 8;
  set ParQData1;

  if visitnum in (60748 60765) then
    do;
      avisit="Before Vaccination 1";
      avisitn=1;
      baseinfo="PRE";
    end;
  else if visitnum in (60750) then
    do;
      avisit="1 Week after Vaccination 1";
      avisitn=2;
      baseinfo="POST";
    end;
  else if visitnum in (60751) then
    do;
      avisit="3 Weeks after Vaccination 1";
      avisitn=3;
      baseinfo="POST";
    end;
  else if visitnum in (60752 1165455) then
    do;
      avisit="1 Week after Vaccination 2";
      avisitn=4;
      baseinfo="POST";
    end;
  else if visitnum in (60753 1165456) then
    do;
      avisit="2 Weeks after Vaccination 2";
      avisitn=5;
      baseinfo="POST";
    end;
  else if visitnum in (60754 60767 1165457) then

```

```

do;
  avisit="1 Month after Vaccination 2";
  avisitn=6;
  baseinfo="POST";
end;
else if visitnum in (60755 60768) then
do;
  avisit="6 Months after Vaccination 2";
  avisitn=7;
  baseinfo="POST";
end;
else if visitnum in (60756 60769) then
do;
  avisit="12 Months after Vaccination 2";
  avisitn=8;
  baseinfo="POST";
end;
else if visitnum in (60757 60770) then
do;
  avisit="24 Months after Vaccination 2";
  avisitn=9;
  baseinfo="POST";
end;

/* Correct AVISIT/AVISITN for 100ug group */
if cohortn=1.16 then
do;

  if visitnum=60752 then
  do;
    avisitn=3.1;
    avisit="4 Weeks after Vaccination 1";
  end;
  else if visitnum=60753 then
  do;
    avisitn=3.2;
    avisit="5 Weeks after Vaccination 1";
  end;
  else if visitnum=60754 then
  do;
    avisitn=3.3;
    avisit="1 Month and 3 weeks after Vaccination 1";
  end;
  else if visitnum=1165454 then
  do;
    avisitn=3.9;
    avisit="Before Vaccination 2";
  end;
end;
label avisitn="Analysis Visit (N)" avisit="Analysis Visit";
run;

/* End visit window */
proc sort data=ParQData1 out=ParQData1;

```

```

    by paramcd usubjid VISITNUM obs;
run;

proc sort data=ParQCond1 out=ParQCond1;
    by paramcd usubjid VISITNUM obs;
run;

data ParQAll1 (drop=obs);
    merge ParQData1 ParQCond1;
    by paramcd usubjid VISITNUM obs;
run;

data AllData;
    set ParQAll1;

    if paramcd='C2NGNT50' then
        paramn=1;
    else if paramcd='C2NGNT90' then
        paramn=2;
    else if paramcd='C19S1IGG' then
        paramn=3;
    else if paramcd='C19RBDIG' then
        paramn=4;
    else if paramcd='C19NIG' then
        paramn=5;
run;

proc sort data=alldata;
    by paramcd;
run;

data bds_apply_window;
    set AllData;
run;

data _parwin _parwinmiss;
    set bds_apply_window;
    _instrec=.;

    if avisitn=. or winmeth=' ' or baseinfo="PRE" then
        do;
            WINCNFL=' ';
            output _parwinmiss;
        end;
    else
        output _parwin;
run;

data _param;
    set _parwin;
    array _num _numeric_;
    array _vis[*] winref;
    array _new[*] visvaval;

```

```

do _i=1 to dim(_vis);
  vis_varname=lowcase(_vis[_i]);

  do _j=1 to dim(_num);
    num_varname=lowcase(vname(_num[_j]));

    if num_varname=vis_varname then
      do;
        _new[_i]=_num[_j];
        leave;
      end;
    end;
  end;
end;
drop vis_varname num_varname _i _j;
run;

data _parscale;
  length method stat $20.;
  set _param;
  _wscale=0;
  _winref=visvaval;
  _winbeg=awlo;
  _winend=awhi;
  _wintarg=awtarget;
  method=trim(upcase(scan(winmeth, 1, '_')));
  stat=trim(upcase(scan(winmeth, 2, '_')));

  if indexw(method, 'F') then
    _Wscale=_winref-_winbeg;
  else if indexw(method, 'L') then
    _Wscale=_winend-_winref;
  else if indexw(method, 'A') then
    _Wscale=1;
  else if indexw(method, 'C') then
    _Wscale=abs(_winref-_wintarg);
  else if indexw(method, 'CB') then
    _Wscale=abs(_wintarg-_winref)+max(_winref>_wintarg,
    0)*(abs(_wintarg-_winbeg)+1);
  else if indexw(method, 'CA') then
    _Wscale=abs(_wintarg-_winref)+max(_winref<_wintarg,
    0)*(abs(_wintarg-_winbeg)+1);
  else if indexw(method, 'CF') then
    _Wscale=abs(_wintarg-_winref)+max(_winref>_wintarg, 0)*0.5;
  else if indexw(method, 'CL') then
    _Wscale=abs(_wintarg-_winref)+max(_winref<_wintarg, 0)*0.5;
run;

proc sql;
  create table _parpick As select *, min(_wscale) as min_wscales, case when
    _wscale=min(_wscale) then 1 else 0 end as _parpick from _parscale where aval
    ne . or ^missing(avalc) group by usubjid, paramcd, APERIOD, avisitn outer
    union corr select *, 0 as _parpick from _parscale where missing(avalc) group
    by usubjid, paramcd, APERIOD, avisitn;
quit;

```

```

proc sort data=_parpick;
  by usubjid paramcd APERIOD avisitn;
run;

proc summary data=_parpick;
  where _parpick eq 1;
  by usubjid paramcd APERIOD avisitn;
  var min_wscales;
  output out=_parpickmin N(min_wscales)=count;
run;

data _parone _parmany;
  merge _parpick(in=a) _parpickmin(in=b keep=usubjid paramcd APERIOD avisitn
    count);
  by usubjid paramcd APERIOD avisitn;

  if a;

  if count > 1 then
    output _parmany;
  else
    do;

      if _parpick=1 then
        WINCNFL='Y';
      else
        WINCNFL=' ';
      output _parone;
    end;
run;

data _parstat _parnostat;
  set _parmany;

  if stat=' ' then
    output _parnostat;
  else
    output _parstat;
run;

proc sort data=_parnostat out=_parnostat;
  by usubjid paramcd APERIOD avisitn _winref ADT ATM;
run;

data _parnostat1 _parnostat2;
  set _parnostat;
  WINCNFL=' ';

  if _parpick=1 then
    output _parnostat1;
  else
    output _parnostat2;
run;

```

```

data _parnostat1;
  set _parnostat1;
  by usubjid paramcd APERIOD avisitn _winref ADT ATM;

  if last.avisitn then
    do;
      _parpick=1;
      WINCNFL='Y';
    end;
  else
    do;
      _parpick=0;
    end;
run;

data _parnostat;
  set _parnostat1 _parnostat2;
  by usubjid paramcd APERIOD avisitn _winref ADT ATM;
run;

data bds_win_calc (drop=min_wscales visval count method stat);
  attrib VISWIN length=8 label='Visit number to use in analysis' XVISWIN
    length=$40 label='Decode of VisWin' ANL01FL length=$1.
    label="Analysis Record Flag 01:Window Record" WINCNFL length=$1.
    label='Visit Window Contributing Record Flag' WINDERFL length=$1.
    label='Derived Visit Window Record Flag' DTYPE length=$20.
    label='Derivation Type';
  set _parnostat _parone _parwinmiss;

  if _parpick=1 then
    do;

      if avisitn ne . then
        do;
          VisWin=avisitn;
          XVisWin=avisitn;
          ANL01FL='Y';
        end;
      else
        ANL01FL=' ';
    end;

  if _instrec=1 then
    WINDERFL='Y';
  else
    WINDERFL=' ';

  if WINDERFL='Y' and index(winmeth, '_') > 0 then
    do;
      DTYPE=scan(winmeth, 2, '_');
      SRCSEQ=.;
      SRCDOM=" ";
      SRCVAR=" ";
    end;

```

```

    end;
run;

/* Create baseline */
data bds_base_dsin(drop=_timefield _dosetime);
    set bds_win_calc;
    length Basetype $100. basec $200;
    Base=.;
    BaseType=" ";
    BnrLo=.;
    BnrHi=.;
    Basec=" ";
    AnrLo=.;
    AnrHi=.;
    format _dataDateTime _trtdatetime datetime18.;
    _prdbasecat=1;

    if ^missing(Tr01Stm) then
        _dosetime=Tr01Stm;
    else
        _dosetime="24:00"t;
        _trtdatetime=dhms(Tr01Sdt, 0, 0, _dosetime);

    if ^missing(atm) then
        _timefield=atm;
    else
        _timefield="00:00"t;
        _dataDateTime=dhms(adtt, 0, 0, _timefield);
        _minSelectableDateTime=_trtdatetime-(99999*24*60*60);
        BasDerFl=" ";
run;

proc sort data=bds_base_dsin;
    by usubjid PARAMCD _datadatetime;
run;

data bds_base_dsin;
    set bds_base_dsin;
    _nflag1=_n_;
run;

data pre_baseline;
    set bds_base_dsin;
    Where (^Missing(Aval) or ^Missing(Avalc)) and (avisitn=1 and avalc ne "" and
        adt<=vax101dt and isuse="Y");
run;

proc sql;
    create table pre_basedatetime as select distinct usubjid, PARAMCD ,
        max(_datadatetime) as _basedatetime from pre_baseline as a group by usubjid,
        PARAMCD order by usubjid, PARAMCD;
quit;

data pre_baseline;

```

```

    set pre_baseline;
    Where (avisitn in (1) and avalc ne "");
run;

proc sql noprint;
    create table getbasetype as select distinct usubjid, PARAMCD, "Study" as
        _basetype length=100 from pre_baseline order by usubjid, PARAMCD;
quit;

proc sort data=pre_baseline;
    by usubjid _prdbasecat PARAMCD descending _datadatetime;
run;

data pre_baseline;
    set pre_baseline;
    by usubjid _prdbasecat PARAMCD descending _datadatetime;
    retain _count_base_obs;

    if first.PARAMCD then
        _count_base_obs=1;
    else
        _count_base_obs=_count_base_obs+1;
run;

data pre_baseline;
    set pre_baseline;
    _nflag2=_nflag1;
run;

proc sort data=pre_baseline;
    by usubjid PARAMCD PARAMCD ADT;
run;

data pre_baseline_first_last;
    set pre_baseline;
    by usubjid PARAMCD;

    if LAST.PARAMCD then
        do;
            _nflag2=_nflag1;
            output pre_baseline_first_last;
        end;
run;

proc sort data=bds_base_dsin;
    by usubjid PARAMCD PARAMCD ADT;
run;

proc sort data=pre_baseline_first_last;
    by usubjid PARAMCD PARAMCD ADT;
run;

data bds_base_dsin(drop=_aval _avalc _AnrLo _AnrHi);
    merge bds_base_dsin(in=a) pre_baseline_first_last(rename=(aval=_aval

```



```

    Avalc=_Avalc AnrLo=_AnrLo AnrHi=_AnrHi) keep=usubjid PARAMCD aval avalc
    _nflag2 AnrLo AnrHi) getbasetype pre_basedatetime;
by usubjid PARAMCD;

if a;
base=_Aval;
basec=put(_Avalc, $200.);
BnrLo=_AnrLo;
BnrHi=_AnrHi;
Basetype=_Basetype;

if _nflag1=_nflag2 then
    do;
        efbascn=1;
        efbasflg=1;
    end;
run;

data bds_base_dsin(drop=_prdbasecat AnrLo BnrLo AnrHi BnrHi);
set bds_base_dsin;
attrib efbascn label='Contributes to Baseline Derivation' efbasflg
    label='Baseline record flag(1=YES) (num)' ABLFL label='Baseline Record Flag'
    APSBLFL Label='Post Baseline Record Flag' Base Label='Baseline Value' BaseC
    Label='Baseline Value (C)' BaseType Label='Baseline Type' BasCnFl
    label='Baseline Contributing Records' BasDerFl
    Label='Baseline Derived Record';

if (^missing(_basedatetime) and _datadatetime>_basedatetime)
    or (missing(_basedatetime) and _datadatetime>_trtdatetime
    and ^missing(_trtdatetime)) then
    APSBLFL="Y";

If EfBasCn=1 then
    BasCnFl="Y";

If Efbasflg=1 then
    do;
        ABLFL="Y";

        if BaseInfo="PRE" and ABLFL="Y" then
            do;
                ANL01FL="Y";
                VisWin=AvisitN;
                XvisWin=Avisit;
            end;
    end;
run;

proc sort data=bds_base_dsin out=baseflag1 nodupkey;
    by usubjid PARAMCD _basedatetime;
    where (^missing(base) or ^missing(basec)) and apsblfl="Y";
run;

proc sort data=bds_base_dsin;

```

```

    by usubjid PARAMCD;
run;

data baseflag2(keep=usubjid PARAMCD);
    merge bds_base_dsin(in=a) baseflag1(in=b drop=_basedatetime);
    by usubjid PARAMCD;

    if a and b;

    if (^missing(aval) or ^missing(avalc)) and (_datadatetime>_basedatetime
        and ^missing(_basedatetime));
run;

proc sort data=baseflag2 nodupkey;
    by usubjid PARAMCD;
run;

data bds_base(drop=_datadatetime _basedatetime _trtdatetime
    _minSelectabledatetime _nflag: _basetype:);
    merge bds_base_dsin(in=a) baseflag2(in=b);
    by usubjid PARAMCD;

    if a and b then
        do;
            Ablpblfl="Y";
        end;
    else if a and not b then
        Ablpblfl="N";
    attrib ABLPBLFL label='Baseline and post-baseline flag';

    if missing(base) and missing(basec) then
        do;
            Ablfl=" ";
            Ablpblfl=" ";
            BaseType=" ";
        end;
run;

/* Change from baseline */
data bds_chg_base;
    set bds_base;

    if BASE>. and AVAL>. then
        do;
            CHG=AVAL - BASE;
        end;
    else
        do;
            CHG=.;
        end;
run;

data bds_chg_base;
    set bds_chg_base;

```

```

if BASE>. and AVAL>. then
  do;

    if BASE ne 0 then
      do;
        PCHG=((AVAL - BASE)/BASE)*100;
      end;
    else
      do;
        PCHG=.;
      end;
    end;
  else
    do;
      PCHG=.;
    end;
  end;
run;

data bds_chg_base;
  attrib CHG label="Change from Baseline" PCHG
    label="Percent Change from Baseline";
  set bds_chg_base;

  if APSBLFL ne 'Y' then
    do;
      CHG=.;
      PCHG=.;
    end;
run;

data bds_base_lis;
  set bds_chg_base;
  where BASE ne . and PCHG=.;
  keep USUBJID PARAMCD VISIT SRCSEQ AVAL BASE CHG;
run;

proc sort data=bds_base_lis;
  by USUBJID PARAMCD VISIT SRCSEQ;
run;

data final;
  set bds_chg_base;
run;

data final;
  attrib ANL02FL length=$1. label="Analysis Record Flag 02:In TRT Period";
  set final;
  _obs=_N_;

  if APERSDT < ADT <=APEREDT then
    ANL02FL='Y';
run;

```

```

/* Create LLOQ */
data final;
  set final;

  if paramcd in ("C2NGNT50" "C2NGNT90") then
    islloq=20;
  else if paramcd in ("C19S1IGG") then
    islloq=1.2665;
  else if paramcd in ("C19RBDIG") then
    islloq=1.1505;
  else if paramcd in ("C19NIG") then
    islloq=.;

  if avalc not in (" " "IND" "QNS" "INDETERMINATE") then
    anl03fl='Y';

  if not missing(islloq) and aval >=islloq then
    anl04fl='Y';
run;

data final;
  length param $ 200 BSSEROC $ 50;
  set final;
  base4f=base;

  /* Impute BLQ values per SAP and create BSSEROC/BSSERON for param other than C19NIG */
  if paramcd not in ("C19NIG") then
    do;

      if (aval < islloq and aval ne .) or avalc in ('BLQ') then
        do;
          aval=islloq*.5;
          avalc=ifc(^missing(aval), strip(put(aval, best.)), "");
        end;

      if (base < islloq and base ne .) or basec in ('BLQ') then
        do;
          base=islloq*.5;
          base4f=islloq;
          basec=ifc(^missing(base), strip(put(base, best.)), "");
          BSSEROC="< LLOQ";
          BSSERON=1;
        end;
      else
        do;
          BSSEROC=">= LLOQ";
          BSSERON=2;
        end;

      if avalc in ("QNS" "IND") then
        do;
          avalc="";
          aval=.;
        end;
    end;

```

```

if chg ne . then
  chg=aval - base;

if base >0 and ablfl ne 'Y' then
  r2base=(aval/base);

if base4f >0 and ablfl ne 'Y' then
  r2base4f=(aval/base4f);
dtype='LLOQIMP';

if missing(base) then
  do;
    BSSEROC="Missing";
    BSSERON=0;
  end;

if not missing(base) then
  basetype=trim(left(basetype))||' ('||trim(left(dtype))||')';
end;

/* Create aval for C19NIG */
else
  do;

    if avalc in ("NEG" "neg") then
      aval=0;
    else if avalc in ("POS" "pos") then
      aval=1;

    if basec in ("NEG" "neg") then
      do;
        base=0;
        base4f=0;
      end;
    else if basec in ("POS" "pos") then
      do;
        base=1;
        base4f=1;
      end;
    dtype="";
  end;

if r2base4f ge 4 then
  fold4fl="Y";
label r2base='Ratio to Baseline' anl03fl='Analysis Flag 03:valid value'
  anl04fl='Analysis Flag 04:ge LLoQ' BSSERON="Baseline serostatus (N)"
  BSSEROC="Baseline serostatus" ISLLOQ="Lower Limit of Quantitation"
  fold4fl="Achieve 4-Fold Rise Flag" base4f="Baseline Value for 4 Fold Rise"
  r2base4f="Ratio to baseline for 4 Fold Rise";

run;

data final1;
  set final;

```

```

    if isuse ne "Y";
run;

proc sort data=final;
    where isuse="Y";
    by usubjid;
run;

proc sort data=final out=v3(keep=usubjid) nodupkey;
    where visitnum in (60754 60767) and isuse="Y";
    by usubjid;
run;

data final;
    merge final v3(in=a);
    by usubjid;
    v3=a;

    if v3=0 and (index(visit, "COVID") or visit in ('V101_VAX3'
    'V201_SURVEIL_CONSENT')) then
        do;

            if 28<=adt-vax102dt<=42 then
                do;
                    avisitn=6;
                    avisit="1 Month after Vaccination 2";
                    ANL01FL="Y";
                end;
            else
                do;
                    avisitn=.;
                    avisit="";
                    ANL01FL="";
                end;
            end;
        else if v3=1 and (index(visit, "COVID") or visit in ('V101_VAX3'
        'V201_SURVEIL_CONSENT')) then
            do;
                avisitn=.;
                avisit="";
                ANL01FL="";
            end;
        run;

data final;
    set final;

    if avisitn>=4 and vax201dt ne . and adt>vax201dt then
        ANL01FL="";
run;

/* Generate GMR */
proc sort data=final out=gmr_pre;

```

```
by usubjid visitnum adt;  
run;
```

```
proc transpose data=gmr_pre out=gmr_pre_t;  
by usubjid visitnum adt;  
id paramcd;  
var aval;  
run;
```

```
data gmr;  
set gmr_pre_t;
```

```
if not missing(C2NGNT50) and not missing(C19S1IGG) then  
do;  
paramn=11;  
paramcd="NT50_S1";
```

```
/* param="SARS-CoV-2-mNG for serum antibody NT50 to COVID-19 S1 IgG"; */  
param="SARS-CoV-2 serum neutralizing titer 50 to COVID-19 S1 IgG";  
aval=round(C2NGNT50/C19S1IGG, 0.0001);  
output;  
end;
```

```
if not missing(C2NGNT90) and not missing(C19S1IGG) then  
do;  
paramn=12;  
paramcd="NT90_S1";
```

```
/* param="SARS-CoV-2-mNG for serum antibody NT90 to COVID-19 S1 IgG"; */  
param="SARS-CoV-2 serum neutralizing titer 90 to COVID-19 S1 IgG";  
aval=round(C2NGNT90/C19S1IGG, 0.0001);  
output;  
end;
```

```
if not missing(C2NGNT50) and not missing(C19RBDIG) then  
do;  
paramn=13;  
paramcd="NT50_RB";
```

```
/* param="SARS-CoV-2-mNG for serum antibody NT50 to COVID-19 RBD Ig"; */  
param="SARS-CoV-2 serum neutralizing titer 50 to COVID-19 RBD Ig";  
aval=round(C2NGNT50/C19RBDIG, 0.0001);  
output;  
end;
```

```
if not missing(C2NGNT90) and not missing(C19RBDIG) then  
do;  
paramn=14;  
paramcd="NT90_RB";
```

```
/* param="SARS-CoV-2-mNG for serum antibody NT90 to COVID-19 RBD Ig"; */  
param="SARS-CoV-2 serum neutralizing titer 90 to COVID-19 RBD Ig";  
aval=round(C2NGNT90/C19RBDIG, 0.0001);  
output;
```

FDA-CBER-2022-5812-0736449

```

    end;
    keep usubjid visitnum adt paramn paramcd param aval;
run;

proc sort data=gmr;
    by usubjid visitnum adt;
run;

proc sort data=gmr_pre out=shell nodupkey;
    by usubjid visitnum adt;
run;

data gmr;
    merge shell(drop=paramn paramcd param aval) gmr(in=a);
    by usubjid visitnum adt;

    if a;
    dtype="Derived";
    avalc=strip(put(aval, best.));
    base=.;
    base4f=.;
    basec="";
    basetype="";
    r2base=.;
    r2base4f=.;
    fold4fl="";
    islloq=.;
    isstresc="";
    srcseq=.;
    anl03fl="";
    anl04fl="";
    SRCDOM="";
    SRCVAR="";
    BSSEROC="";
    BSSERON=.;
    ISTSTDTL="";
run;

/* Add BASE/R2BASE for GMR */
proc sort data=gmr;
    by subjid paramcd adt;
run;

data gmr;
    set gmr(drop=base base4f basec basetype r2base r2base4f ABLPBLFL);
    by subjid paramcd adt;
    retain base;

    if first.paramcd and ABLFL="Y" then
        base=aval;
    else if first.paramcd then
        base=.;
    else if base ne . then
        base+0;

```



```

if not missing(base) then
  do;
    basec=strip(put(base, best.));
    basetype="Study (Derived)";
  end;

if base >0 and ablfl ne 'Y' then
  r2base=(aval/base);
run;

data base;
  set gmr;
  where ABLFL="Y";

proc sort nodupkey;
  by subjid;
run;

data post;
  set gmr;
  where ABLFL^="Y";

proc sort nodupkey;
  by subjid;
run;

data both;
  merge base(in=a) post(in=b);
  by subjid;

  if a and b;
  ABLPBLFL="Y";
run;

data gmr;
  merge gmr(in=a) both(keep=subjid ABLPBLFL in=b);
  by subjid;

  if a;

  if not b then
    ABLPBLFL="N";
run;

data Nbind(drop=ABLPBLFL) final;
  set final gmr;

  if param="N-binding antibody - N-binding Antibody Assay" then
    output Nbind;
  else
    output final;
run;

```

```

data base;
  set Nbind;
  where ABLFL="Y";

proc sort nodupkey;
  by subjid;
run;

data post;
  set Nbind;
  where ABLFL^="Y";

proc sort nodupkey;
  by subjid;
run;

data both;
  merge base(in=a) post(in=b);
  by subjid;

  if a and b;
  ABLPBLFL="Y";
run;

data Nbind;
  merge Nbind(in=a) both(keep=subjid ABLPBLFL);
  by subjid;

  if a;

  if ABLPBLFL="" then
    ABLPBLFL="N";
run;

data final;
  set final Nbind final1;
run;

data final;
  set final;

  /* Re-create ADY per each vax date */
  if not missing(vax101dt) and adt>=vax101dt then
    ady=adt-vax101dt+1;

  if not missing(vax102dt) and adt>vax102dt then
    ady=adt-vax102dt+1;

  if missing(vax101dt) then
    ady=.;

  if cohortn=1.16 then
    do;

```

```

if visitnum=60752 then
  do;
    avisitn=3.1;
    avisit="4 Weeks after Vaccination 1";
    anl01fl="Y";
  end;
else if visitnum=60753 then
  do;
    avisitn=3.2;
    avisit="5 Weeks after Vaccination 1";
    anl01fl="Y";
  end;
else if visitnum=60754 then
  do;
    avisitn=3.3;
    avisit="1 Month and 3 weeks after Vaccination 1";
    anl01fl="Y";
  end;
else if visitnum=1165454 then
  do;
    avisitn=3.9;
    avisit="Before Vaccination 2";
    anl01fl="Y";
  end;
end;

```

/* Create all-available and evaluable flags */

```

if phasen in (1) then
  do;

    if avisitn<4 then
      do;

        if eval01fl="Y" then
          EVIMMFL="Y";
        else
          EVIMMFL="N";

        if aai01fl="Y" then
          AAIMMFL="Y";
        else
          AAIMMFL="N";
      end;
    else if avisitn>=4 then
      do;

        if eval02fl="Y" then
          EVIMMFL="Y";
        else
          EVIMMFL="N";

        if aai02fl="Y" then
          AAIMMFL="Y";
        else

```

```

        AAIMMFL="N";
    end;
end;

if phasen in (2 3 4) then
    do;

        if eval02fl="Y" then
            EVIMMFL="Y";
        else
            EVIMMFL="N";

        if aai02fl="Y" then
            AAIMMFL="Y";
        else
            AAIMMFL="N";
        end;

    if not missing(PD1POSDT) and adt ge PD1POSDT then
        EVIMMFL="N";
    label EVIMMFL="Evaluable Immunogenicity Record Flag"
        AAIMMFL="All-available Immunogenicity Record Flag";
run;

proc sort data=final out=base(keep=usubjid paramcd) nodupkey;
    where ablfl="Y";
    by usubjid paramcd;
run;

proc sort data=final out=post(keep=usubjid paramcd) nodupkey;
    where Apsblfl="Y";
    by usubjid paramcd;
run;

data bp(keep=usubjid paramcd);
    merge base(in=a) post(in=b);
    by usubjid paramcd;

    if a and b;
run;

proc sort data=final;
    by usubjid paramcd;
run;

data final;
    merge final(in=a) bp(in=b);
    by usubjid paramcd;

    if a;
    Ablpblfl="N";

    if b then
        Ablpblfl="Y";

```

```

run;

data final;
  set final;

  if paramn not in (13, 14);
  isdtc=strip(scan(isdtc, 1, "T"));

  if paramcd="C19NIG" then
    do;
      avisitn=.;
      avisit="";
      anl01fl=" ";
    end;

  if isuse="N" then
    do;
      avisitn=.;
      avisit="";
      anl01fl=" ";
    end;

run;

proc sort data=final;
  by usubjid param aperiod AVISITN ADT visitnum DTYPE;
run;

data final_reset;
  set final;

  if ^missing(DTYPE) and upcase(dtype) not in ('LLOQIMP', 'DERIVED') then
    do;

      if (^missing(DTYPE) and upcase(dtype) not in ('LLOQIMP', 'DERIVED') ) then
        ADT=.;
        ATM=.;
        ADTM=.;
        ATMF=" ";

      if (^missing(DTYPE) and upcase(dtype) not in ('LLOQIMP', 'DERIVED') ) then
        ADY=.;
        SRCSEQ=.;

      if (^missing(DTYPE) and upcase(dtype) not in ('LLOQIMP', 'DERIVED') ) then
        APERDY=.;
    end;

run;

/* Output ADVA */
data datvprot.adva (label="Immunogenicity Analysis Dataset");
  retain STUDYID USUBJID SUBJID SITEID TRTP TRTPN TRTA TRTAN ISDTC ADT ADY
  AVISIT AVISITN VISIT VISITNUM PARCAT1 PARCAT1N PARAM PARAMN PARAMCD AVAL
  AVALC BASE BASEC BASETYPE ABLFL APSBLFL ABLPBLFL DTYPE R2BASE SRCDOM SRCVAR
  SRCSEQ ANL01FL ANL03FL ANL04FL SRCVAR SRCDOM SRCSEQ ISLLOQ ISSTRESC EVIMMFL

```

FDA-CBER-2022-5812-0736455

```

AAIMMFL BSSERON BSSEROC APERIOD APERIODC APERSDTM APEREDTM BASE4F R2BASE4F
FOLD4FL ISTSTDITL USUBJID SUBJID SITEID AGE AGEU SEX SEXN RACE RACEN RANDFL
SAFFL ARM ARMCD ACTARM ACTARMCD TRTSDT TRTEDT TRTSTM TRTETM TRT01A TRT01AN
TRT02A TRT02AN TRT01P TRT01PN TRT02P TRT02PN TR01SDT TR01STM TR01SDTM TR01EDT
TR01ETM TR01EDTM TR02SDT TR02STM TR02SDTM TR02EDT TR02ETM TR02EDTM COHORTN
COHORT VAX101DT VAX102DT VAX10UDT VAX201DT VAX202DT SAFFL AAI01FL EVAL02FL
AAI02FL EVAL01FL DOSALVL DOSALVLN DOSPLVL DOSPLVLN AGEGR1 AGEGR1N AGEGR2
AGEGR2N AGEGR4 AGEGR4N PHASE PHASEN COVBLST HIVFL PEDIMMFL EV1MD2FL AGETR01
VAX101 VAX102 VAX10U VAX201 VAX202 VAX20U VAX20UDT UNBLNDDT MULENRFL RAND1FL
TRTSDTM TRTEDTM;

```

```

set final_reset;

```

```

keep STUDYID USUBJID SUBJID SITEID TRTP TRTPN TRTA TRTAN ISDTC ADT ADY AVISIT
AVISITN VISIT VISITNUM PARCAT1 PARCAT1N PARAM PARAMN PARAMCD AVAL AVALC BASE
BASEC BASETYPE ABLFL APSBLFL ABLPBLFL DTYPE R2BASE SRCDOM SRCVAR SRCSEQ
ANL01FL ANL03FL ANL04FL SRCVAR SRCDOM SRCSEQ ISLLOQ ISSTRESC EVIMMFL AAIMMFL
BSSERON BSSEROC APERIOD APERIODC APERSDTM APEREDTM BASE4F R2BASE4F FOLD4FL
ISTSTDITL USUBJID SUBJID SITEID AGE AGEU SEX SEXN RACE RACEN RANDFL SAFFL ARM
ARMCD ACTARM ACTARMCD TRTSDT TRTEDT TRTSTM TRTETM TRT01A TRT01AN TRT02A
TRT02AN TRT01P TRT01PN TRT02P TRT02PN TR01SDT TR01STM TR01SDTM TR01EDT
TR01ETM TR01EDTM TR02SDT TR02STM TR02SDTM TR02EDT TR02ETM TR02EDTM COHORTN
COHORT VAX101DT VAX102DT VAX10UDT VAX201DT VAX202DT SAFFL AAI01FL EVAL02FL
AAI02FL EVAL01FL DOSALVL DOSALVLN DOSPLVL DOSPLVLN AGEGR1 AGEGR1N AGEGR2
AGEGR2N AGEGR4 AGEGR4N PHASE PHASEN COVBLST HIVFL PEDIMMFL EV1MD2FL AGETR01
VAX101 VAX102 VAX10U VAX201 VAX202 VAX20U VAX20UDT UNBLNDDT MULENRFL RAND1FL
TRTSDTM TRTEDTM;

```

```

if ^missing(dtype) and (WINDERFL='Y' or BASDERFL='Y') then
do;

```

```

    if ^missing(avaldec) then

```

```

    do;

```

```

        AVAL=round(AVAL, (0.1)**AVALDEC);

```

```

        AVALC=strip(put(AVAL, best.));

```

```

    end;

```

```

end;

```

```

label ADY="Analysis Relative Day" AVAL="Analysis Value"

```

```

AVALC="Analysis Value (C)" DTYPE="Derivation Type" PARAM="Parameter"

```

```

PARAMCD="Parameter Code" PARAMN="Parameter (N)"

```

```

PARCAT1="Parameter Category 1" PARCAT1N="Parameter Category 1 (N)"

```

```

SRCVAR="Source Variable" SRCDOM="Source Data" SRCSEQ="Source Sequence Number";

```

```

run;

```

```

proc printto;

```

```

run;

```