

```

/* *****/
** Program Name   : adfacevd.sas                **
** Date Created  : 10Mar2021                    **
** Programmer Name : (b) (6)                    **
** Purpose       : Create adfacevd dataset      **
** Input data    : face vs ex supface supvvs adsl **
** Output data   : adfacevd.sas7bdat           **
*****/

```

```

options mprint mlogic symbolgen mprint symbolgen mlogic nocenter missing=" ";
proc datasets library=WORK kill nolist nodetails;
quit;

```

```

**Setup the environment**

```

```

%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/adfacevd.rpt"
log="&prot./analysis/esub/logs/adfacevd.log" new;
run;

```

```

*****,
*Specification 1. Create foramts                *,
*****,

```

```

Proc format;

```

```

  invalue Sevgrade
    "NONE" = 0
    "MILD" = 1
    "MODERATE" = 2
    "SEVERE" = 3
  ;
  invalue FTEMCAT
    "<38.0 C" = 0
    "38.0 C to 38.4 C" = 1
    ">38.4 C to 38.9 C" = 2
    ">38.9 C to 40.0 C" = 3
    ">40.0 C" = 4
  ;
  value FTEMCAT
    0="<38.0 C"
    1="38.0 C to 38.4 C"
    2=">38.4 C to 38.9 C"
    3=">38.9 C to 40.0 C"
    4=">40.0 C"
  ;
  invalue AVALCAT
    ">0-2.0" = 0
    ">2.0-5.0" = 1
    ">5.0-10.0" = 2
    ">10.0" = 3

```

```

;
run;

*****
*Specification 2. Setup Lookup Meta file for parameters      *;
*****
data _lookup;
    length fatestd $8. paramcd $8. fatest $40. faobj $100. flagdl $1.;

** Occurrence Indicator*;
FATESTCD='OCCUR' ;
FATEST='Occurrence Indicator';
    faobj='REDNESS' ;paramn=7;paramcd='OCISR';OUTPUT;
    faobj='SWELLING' ;paramn=36;paramcd='OCINS';OUTPUT;
    faobj='PAIN AT INJECTION SITE' ;paramn=50;paramcd='OCPIS';output;
    faobj='FEVER' ;paramn=71;paramcd='OCFEVER';output;
    faobj='FATIGUE' ;paramn=82;paramcd='OCFATIG';output;
    faobj='HEADACHE' ;paramn=92;paramcd='OCHEAD';output;
    faobj='CHILLS' ;paramn=102;paramcd='OCCHILLS';output;
    faobj='DIARRHEA' ;paramn=270;paramcd='OCDIAR';output;
    faobj='MUSCLE PAIN' ;paramn=310;paramcd='OCMPNIS';output;
    faobj='JOINT PAIN' ;paramn=390;paramcd='OCJOPAIN';output;
    faobj='VOMITING' ;paramn=400;paramcd='OCVOMI';OUTPUT;

    faobj='HOSPITALIZED FOR DIARRHEA' ;paramn=410;paramcd='OCHODI';OUTPUT;
    faobj='HOSPITALIZED FOR HEADACHE' ;paramn=415;paramcd='OCHOHE';output;
    faobj='HOSPITALIZED FOR INJECTION SITE PAIN' ;paramn=420;paramcd='OCHIS';output;
    faobj='HOSPITALIZED FOR JOINT PAIN' ;paramn=425;paramcd='OCHOJP';output;
    faobj='HOSPITALIZED FOR NEW OR WORSENERED JOINT PAIN' ;paramn=425;paramcd='OCHOJP';output;
    faobj='HOSPITALIZED FOR TIREDNESS (FATIGUE)' ;paramn=430;paramcd='OCHOFA';output;
    faobj='HOSPITALIZED FOR VOMITING' ;paramn=440;paramcd='OCHOVO';output;
    faobj='HOSPITALIZED FOR MUSCLE PAIN' ;paramn=445;paramcd='OCHOMP';output;
    faobj='HOSPITALIZED FOR NEW OR WORSENERED MUSCLE PAIN' ;paramn=445;paramcd='OCHOMP';output;
    faobj='HOSPITALIZED FOR CHILLS' ;paramn=455;paramcd='OCHOCHIL';output;

    faobj='HOSPITALIZATION FOR DIARRHEA' ;paramn=410;paramcd='OCHODI';OUTPUT;
    faobj='HOSPITALIZATION FOR HEADACHE' ;paramn=415;paramcd='OCHOHE';output;
    faobj='HOSPITALIZATION FOR INJECTION SITE PAIN' ;paramn=420;paramcd='OCHIS';output;
    faobj='HOSPITALIZATION FOR JOINT PAIN' ;paramn=425;paramcd='OCHOJP';output;
    faobj='HOSPITALIZATION FOR NEW OR WORSENERED JOINT PAIN' ;paramn=425;paramcd='OCHOJP';output;
    faobj='HOSPITALIZATION FOR TIREDNESS (FATIGUE)' ;paramn=430;paramcd='OCHOFA';output;
    faobj='HOSPITALIZATION FOR VOMITING' ;paramn=440;paramcd='OCHOVO';output;
    faobj='HOSPITALIZATION FOR MUSCLE PAIN' ;paramn=445;paramcd='OCHOMP';output;
    faobj='HOSPITALIZATION FOR NEW OR WORSENERED MUSCLE PAIN'
;paramn=445;paramcd='OCHOMP';output;
    faobj='HOSPITALIZATION FOR CHILLS' ;paramn=455;paramcd='OCHOCHIL';output;

** Diameter for REDNESS and SWELLING*;
FATEST='Diameter';
FATESTCD='DIAMETER' ;
    faobj='REDNESS' ;paramn=4;paramcd='DIARE';output;
    faobj='SWELLING' ;paramn=33; paramcd='DIASW';output;

** Maximum Diameter*;

```

```
FATESTCD='MAXDIAM' ;
FATEST='Maximum Diameter';
faobj='REDNESS' ;paramn=6;paramcd='MDIRE';output;
faobj='SWELLING' ;paramn=35; paramcd='MDISW';output;
```

```
** Minimum Diameter*;
FATESTCD='MINDIAM' ;
FATEST='Minimum Diameter';
faobj='REDNESS' ;paramn=5;paramcd='MIDRE';output;
faobj='SWELLING' ;paramn=34; paramcd='MIDSW';output;
```

```
** Severity*;
FATEST='Severity/Intensity';
fatestcd='SEV' ;
faobj='REDNESS' ;paramn=1;paramcd='SEVREDN';output;
faobj='SWELLING' ;paramn=30;paramcd='SEVSWEL';output;
faobj='PAIN AT INJECTION SITE' ;paramn=51;paramcd='SEVPIS';output;
faobj='FATIGUE' ;paramn=80;paramcd='SEVFATI';output;
faobj='HEADACHE' ;paramn=90;paramcd='SEVHEAD';output;
faobj='CHILLS' ;paramn=100;paramcd='SEVCHIL';output;
faobj='DIARRHEA' ;paramn=271;paramcd='SEVDIAR';output;
faobj='MUSCLE PAIN' ;paramn=311;paramcd='SEVMUSP';output;
faobj='JOINT PAIN' ;paramn=391;paramcd='SEVJOIN';output;
faobj='VOMITING' ;paramn=401;paramcd='SEVVOMI';output;
```

```
** Grade 4 criteria*;
FATEST='Grade 4 Criteria Met';
fatestcd='G4CRIMET' ;
faobj='REDNESS' ;paramn=3;paramcd='G4CRR';output;
faobj='SWELLING' ;paramn=32;paramcd='G4CRS';output;
```

```
** Maximum Severity*;
FATEST='Maximum Severity';
fatestcd='MAXSEV' ;
faobj='REDNESS' ;paramn=2;paramcd='MSERE';output;
faobj='SWELLING' ;paramn=31;paramcd='MSESW';output;
faobj='PAIN AT INJECTION SITE' ;paramn=52;paramcd='MSPIS';output;
faobj='FATIGUE' ;paramn=81;paramcd='MAXSFAT';output;
faobj='HEADACHE' ;paramn=91;paramcd='MAXSHEA';output;
faobj='CHILLS' ;paramn=101;paramcd='MAXCHIL';output;
faobj='DIARRHEA' ;paramn=272;paramcd='MAXDIAR';output;
faobj='MUSCLE PAIN' ;paramn=312;paramcd='MAXSMP';output;
faobj='JOINT PAIN' ;paramn=392;paramcd='MAXSJP';output;
faobj='VOMITING' ;paramn=402;paramcd='MAXSVOM';output;
```

```
**Maximum Temperature and Medications*;
fatest='Maximum Temperature'; fatestcd='MAXTEMP' ;FAOBJ='FEVER';paramn=70;paramcd=fatestcd;output;
fatest='Medication to Treat Fever or Pain'; fatestcd='MEDTFVPN'
;FAOBJ='MEDICATIONS';paramn=182;paramcd=fatestcd;output;
fatest='Stop Date Meds Given to Trt/Pnt Symptoms'; fatestcd='STPDMEDP'
;FAOBJ='MEDICATIONS';paramn=189;paramcd=fatestcd;output;
run;
data lookup;
set _lookup;
```

```

** Derive PARAM;
if fatestd in ('DIAMETER' 'MINDIAM') or paramcd in ('MADRE' 'MADSW') then param=strip(propcase(faobj))||
' strip(lowcase(fatest))||' cm';
else if fatestd in ('MEDTFVPN') then param=strip(propcase(faobj))||' ||propcase(FATEST);
else PARAM=strip(propcase(faobj))||' ||strip(lowcase(fatest));
run;

*****;
*Specification 3.1 Input source FACE/VS SDTMs *;
*****;

data facevd;
    set dataprot.face;
    where facat in ("REACTOGENICITY EVENT", "REACTOGENICITY", "REACTOGENICITY - UNPLANNED
ASSESSMENT");
run;

proc sql noprint;
    create table domseq as select distinct min(FASEQ) as _minseq, max(FASEQ) as _maxseq from facevd;
quit;

data supfacevd;
    set dataprot.supface;
    do i=1 to n;
        set domseq point=i nobs=n;
        output;
    end;
run;

data supfacevd(drop=_minseq _maxseq);
    set supfacevd;
    where (strip(uppercase(idvar))="FASEQ" and _minseq<=input(idvarval, ? best.)<=_maxseq) or
(strip(uppercase(idvar))="FACAT"
and idvarval in ("REACTOGENICITY", "REACTOGENICITY - UNPLANNED ASSESSMENT")) or
(missing(idvar));
run;

data _spmdel_supp_dsin_subset;
    set supfacevd;
run;

data _spmdel_sdtm_ds;
    set facevd;
run;

data _spmdel_supp_dsin_subset_idvar1;
    set _spmdel_supp_dsin_subset;
    where idvar="FASEQ";
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;

```

FDA-CBER-2022-5812-0735957

```

by studyid usubjid idvar idvarval;
id qnam;
idlabel qlabel;
var qval;
quit;

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

data _spmdel_suppsds1 (drop=idvar idvarval _NAME_ _LABEL_);
  set _spmdel_supp_dsin_idvar1_h;

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

*****;
*Specification 3.2 Merge SDTMs and supplemental datasets      *;
*****;

*Merge FACE and SUPPFACE together*;
proc sort data=_spmdel_sdtm_ds out=_ds1;
  by STUDYID USUBJID FASEQ;
run;

proc sort data=_spmdel_suppsds1 out=_ds2;
  by STUDYID USUBJID FASEQ;
run;

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

data facevd;
  set _spmdel_sdtm_temp_out1;
run;

proc datasets library=work noprint;
  delete _spmdel.;
quit;

*Merge VS and SUPPVS together*;
data vsvd;
  set dataprot.vs;
  where vscat in ("REACTOGENICITY", "REACTOGENICITY EVENT",
    "REACTOGENICITY - UNPLANNED TEMPERATURE", "UNPLANNED TEMPERATURE");
run;

proc sql noprint;
  create table domseq as select distinct min(VSSEQ) as _minseq, max(VSSEQ) as _maxseq from vsvd;
quit;

```

```

data suppvsvd;
  set dataprot.suppvsv;
  do i=1 to n;
    set domseq point=i nobs=n;
    output;
  end;
run;

data suppvsvd(drop=_minseq _maxseq);
  set suppvsvd;
  where (strip(upcase(idvar))="VSSEQ" and _minseq<=input(idvarval,? best.)<=_maxseq) or
(strip(upcase(idvar))="VSCAT" and
  idvarval in ("REACTOGENICITY", "REACTOGENICITY - UNPLANNED TEMPERATURE")) or
(missing(idvar));
run;

data _spmdel_supp_dsin_subset;
  set suppvsvd;
run;

data _spmdel_sdtm_ds;
  set vsvd;
run;

data _spmdel_supp_dsin_subset_idvar1;
  set _spmdel_supp_dsin_subset;
  where idvar="VSSEQ";
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=VSSEQ);
  set _spmdel_sdtm_ds;
run;

data _spmdel_suppsds1 (drop=idvar idvarval _NAME_ _LABEL_);
  set _spmdel_supp_dsin_idvar1_h;
  if idvar="VSSEQ";
  VSSEQ=input(idvarval, best12.);
run;

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

```

```
proc sort data=_spmdel_suplds1 out=_ds2;
  by STUDYID USUBJID VSSEQ;
run;
```

```
data _spmdel_sdtm_temp_out1;
  merge _ds1(in=d1) _ds2(in=d2);
  by STUDYID USUBJID VSSEQ;
  if d1;
run;
```

```
data vsvd;
  set _spmdel_sdtm_temp_out1;
run;
proc datasets library=work noprint;
  delete _spmdel;;
quit;
```

```
*****;
*Specification 3.3 Pick up informtaion of FEVER from VS dataset *;
*****;
```

```
proc sql;
  create table vs_fever as select STUDYID, DOMAIN, VSSEQ as FASEQ, USUBJID, VSLNKGRP as FALNKGRP,
  VSLNKID as FALNKID,
  VSCAT as FACAT, VSSTRESN, VSSTRESU, VSORRES, VSORRESU, VSTPTREF as FATPTREF, VSTPT as
  FATPT, VSDTC as FADTC, VSDY
  as FADY, VSSTAT as FASTAT, VSREASND as FAREASND, VISIT, VISITNUM, CLTYP, VSEVAL as
  FAEVAL, VSEVINTX as FAEVINTX
  from vsvd where upcase(VSTESTCD)='TEMP' and (FAEVAL='STUDY SUBJECT' or
  (FAEVAL='INVESTIGATOR' and visit eq ''))
  order by USUBJID, VSTPTREF, VSTPT;
quit;
```

```
proc sql;
  create table vs_fever2 as select a.*, b.FATPTREF as _FATPTREF from (select * from vs_fever where FAEVAL eq
  'INVESTIGATOR') as a
  left join (select distinct USUBJID, FATPTREF, scan(FADTC, 1, 'T') as _fadtc from facevd) as b
  on a.USUBJID=b.USUBJID and a.FADTC=b._FADTC;
quit;
```

```
data facevd;
  set facevd vs_fever(where=(FAEVAL eq 'STUDY SUBJECT') in=a)
  vs_fever2(where=(FAEVAL eq 'INVESTIGATOR' and _FATPTREF ne '') in=b);
```

```
if b then do;
  FACAT="REACTOGENICITY - UNPLANNED ASSESSMENT";
  FATPTREF=strip(_FATPTREF);
  FALNKGRP=strip(FATPTREF)||"-FEVER";
  FALNKID=upcase(compbl(catx('-', FATPTREF, "FEVER")));
end;
```

```
if a or b then do;
  FASCAT="SYSTEMIC";
  FAOBJ="FEVER";
```

```

FAGRPID=upcase(compbl(catx('-', FATPTREF, "SYSTEMIC")));
FATESTCD="OCCUR";
FATEST="Occurrence Indicator";

if .<vsstresn<38 then FASTRESC="N";
else if vsstresn>=38 then FASTRESC="Y";
else FASTRESC=" ";
*Exclude temperatures >42 from analysis*;
if vsstresn>42 then FASTRESC=" ";
end;
run;

data facevd;
set facevd;
FADIR="";
FALNKID=upcase(FALNKID);
FALNKGRP=upcase(FALNKGRP);
run;

**Apply cutoff to add flag after unblinding date**;
proc sort data=facevd;
by usubjid;
run;

proc sort data=vsvd;
by usubjid;
run;

data UNBLNDDT;
set datvprot.adsl;
where UNBLNDDT ne .;
keep usubjid UNBLNDDT;
run;

proc sort;
by usubjid;
run;

data facevd;
merge facevd(in=a) UNBLNDDT(in=b);
by usubjid;

if a and (input(scan(fadtc, 1, "T"), ??yymmdd10.)>=UNBLNDDT>.) then
CUTUNBFL="Y";
drop UNBLNDDT;
run;

data vsvd;
merge vsvd(in=a) UNBLNDDT(in=b);
by usubjid;

if a and (input(scan(vsdtc, 1, "T"), ??yymmdd10.)>=UNBLNDDT>.) then
CUTUNBFL="Y";

```



```

drop UNBLNDDT;
run;

*****
*Specification 3.4 Merge e-diary data with EX dataset *;
*****

data ex(rename=(exlnkgrp=fatptref exloc=faloc exlat=falat exdir=fadir));
  set dataprot.ex;
  exdir="";
run;

data facevd1 facevd2 facevd3;
  set facevd;
  if upcase(FASCAT)='ADMINISTRATION SITE' then output facevd1;
  else if upcase(FASCAT) in ('SYSTEMIC', 'MEDICATIONS GIVEN') then output facevd2;
  else output facevd3;
run;

proc sort data=facevd1;
  by usubjid fatptref;
quit;

proc sort data=ex;
  by usubjid fatptref;
quit;

data facevd111;
  merge facevd1(in=a) ex(keep=usubjid exdose extrt exdosu exstdtc exendtc fatptref);
  by usubjid fatptref;
  if a;
run;

proc sort data=facevd2;
  by usubjid fatptref;
run;

data facevd2;
  merge facevd2(in=a) ex(keep=usubjid exdose extrt exdosu exstdtc exendtc fatptref);
  by usubjid fatptref;
  if a;
run;

data facevd(rename=(fatptref=atptref facat=parcat1 fascat=parcat2 visit=avisit visitnum=avisitn fatpt=atpt fady=ady
  domain=srdom fastresn=aval fastresc=avalc));
  set facevd111 facevd2 facevd3;
  faobj=upcase(faobj);
run;

*****
*Specification 3.5 Include unplanned readings within 7 days after each dose *;
*****

```

```

data facevd face_beyond7;
  set facevd;
  if upcase(avalc)='POTENTIALLY LIFE THREATENING' then avalc="GRADE 4";

  if atpt eq " and FAEVAL='INVESTIGATOR' and faobj ne "MEDICATIONS" then do;
    _fatpt=input(scan(fadtc, 1, 'T'), yymmdd10.)-input(scan(exstdtc, 1, 'T'), yymmdd10.)+1;
    atpt="DAY "||strip(_fatpt);
  end;

  if _fatpt>7 and FAEVAL='INVESTIGATOR' and faobj ne "MEDICATIONS" then output face_beyond7;
  else output facevd;
run;

```

```

data facevd;
  set facevd;
  ADT=input(FADTC, ??is8601da.);
  format ADT date9.;
  ATM=.;
  if length(strip(FADTC))>12 then do;
    ATM=input(substr(FADTC, 12), ??is8601tm.);
  end;
  format ATM time8.;

```

```

  ATPTN=input(substr(ATPT, 4), 8.);
run;

```

```

proc sql;
  create table _temp as select distinct USUBJID, FATESTCD, FAOBJ, ATPTREF, EXTRT, max(ATPTN) as
last_atptn from facevd
  where FASTAT="" group by USUBJID, FATESTCD, FAOBJ, ATPTREF, EXTRT;
  create table facevd as select * from facevd a left join _temp b
  on a.USUBJID=b.USUBJID and a.FATESTCD=b.FATESTCD and a.FAOBJ=b.FAOBJ and
a.ATPTREF=b.ATPTREF and a.EXTRT=b.EXTRT;
quit;

```

```

data facevd1 facevd2;
  set facevd;
  if upcase(FAOBJ)='FEVER' then output facevd1;
  else output facevd2;
run;

```

```

*Derive FTEMCAT/FTEMCATN for fever*;
data facevd1;
  length FTEMCAT FTEMCATC $200. FTEMCATN 8.;
  set facevd1;
  VSORRESN=input(VSORRES, best.);

```

```

  if vsorresu="F" then do;
    If .<vsorresn < 100.4 then FTEMCAT='<38.0 C';
    Else If 100.4<=vsorresn <=101.1 then FTEMCAT='38.0 C to 38.4 C';
    Else If 101.2<=vsorresn <=102.0 then FTEMCAT='>38.4 C to 38.9 C';
    Else If 102.1<=vsorresn <=104.0 then FTEMCAT='>38.9 C to 40.0 C';
    Else If vsorresn > 104.0 then FTEMCAT='>40.0 C';
  end;

```

```

        ftemcatn=input(ftemcat, ?? ftemcat.);
end;
else if vsorresu="C" then do;
    If .<vsorresn < 38 then FTEMCA TC='<38.0 C';
    Else If 38<=vsorresn <=38.4 then FTEMCA TC='38.0 C to 38.4 C';
    Else If 38.4< vsorresn <=38.9 then FTEMCA TC='>38.4 C to 38.9 C';
    Else If 38.9< vsorresn <=40 then FTEMCA TC='>38.9 C to 40.0 C';
    Else If vsorresn > 40 then FTEMCA TC='>40.0 C';
    FTEMCA T=strip(FTEMCA TC);
    ftemcatn=input(ftemcat, ?? ftemcat.);
end;

if missing(VSSTRESN) and fastat eq " then FTEMCA T='Missing';
*Exclude temperatures >42 from analysis*;
if (vsstresu='C' and vsstresn>42) then do; FTEMCA T='Missing'; FTEMCA TN=.;end;
run;

```

```

data facevd_org;
    set facevd1 facevd2;
run;

```

```

*****;
*Specification 4 Local Reaction and Pain medication *;
*****;

```

```

data facevd;
    set facevd_org;
    if upcase(parcat2) NE "SYSTEMIC";
run;

```

```

data _facevd;
    set facevd;
    if upcase(faobj) in ('REDNESS', 'SWELLING') and FATESTCD in ('DIAMETER', 'MAXDIAM');
    if (0 < ATPTN <=7 or (missing(ATPTN) or upcase(PARCAT1)='REACTOGENICITY - UNPLANNED
ASSESSMENT'));
run;

```

```

*****;
*Specification 4.1 Derive SEV/MAXDIAM/MAXSEV for Redness/Swelling *;
*****;

```

```

proc sql;
    create table temp as select a.usubjid, a.faobj, a.atptref, fagrpid, a.falnkgp, parcat1, parcat2, studyid, srdom,
        a.visit, a.visitn, EXDOSE, ExDOSU, EXSTDTC, EXENDTC, extrt, atpt, atptn, a.aval,
        'MAXDIAM' as FATESTCD, 'Maximum Diameter' as FATEST, 'MAXIMUM' as DTYPE from _facevd a ,
        (select usubjid, faobj, atptref, falnkgp, max(aval) as AVAL from _facevd group by usubjid, faobj, atptref,
        falnkgp) b
        where a.usubjid=b.usubjid and a.faobj=b.faobj and a.atptref=b.atptref and a.falnkgp=b.falnkgp and a.aval=b.aval;
quit;

```

```

proc sort data=temp;
    where aval>.;
    by usubjid faobj atptref parcat1;
run;

```

```

proc sort data=temp nodupkey;
    by usubjid faobj atptref;
run;

data temp;
    set temp;

    if upcase(PARCAT1)='REACTOGENICITY - UNPLANNED ASSESSMENT' and
upcase(FATESTCD)='MAXDIAM' then do;
        _unplan='Y';
    end;
    FASTINT='P0H';
    FAENINT="P7D";
    ADY=.;
run;

proc sort data=_facevd;
    by usubjid faobj atptref;
run;

data _facevd;
    set _facevd temp(drop=atpt atptn);
run;

proc sql;
    create table cntmiss as select usubjid, faobj, fatestcd, atptref, count(1) as _cntmiss from _facevd
        where missing(aval) and fatestcd ne 'MAXDIAM' group by usubjid, faobj, atptref;
quit;

proc sql;
    create table cntmisszero as select usubjid, faobj, fatestcd, atptref, count(1) as _cntmisszero from _facevd
        where (missing(aval) or aval=0) and fatestcd ne 'MAXDIAM' group by usubjid, faobj, atptref;
quit;

proc sort data=_facevd;
    by usubjid faobj atptref;
run;

proc sort data=cntmiss;
    by usubjid faobj atptref;
run;

proc sort data=cntmisszero;
    by usubjid faobj atptref;
run;

data _facevd;
    merge _facevd(in=a) cntmiss(drop=fatestcd) cntmisszero(drop=fatestcd);
    by usubjid faobj atptref;
    if a;

    if fatestcd eq 'MAXDIAM' then do;
        if _cntmiss=7 then aval=.;

```

```

        else if _cntmisszero=7 then aval=0;
        if AVAL=0 then AVALC='NONE';
        if missing(AVAL) then AVALC=' ';
    end;
run;

data _facevd1;
    set _facevd;
    where FAEVAL ne "INVESTIGATOR";
    length sevgrade $10.;

    if FATESTCD='DIAMETER' then do;
        FATESTCD='SEV';
        FATEST='Severity/Intensity';
        FASTRESU=' ';
    end;
    if FATESTCD='MAXDIAM' then do;
        FATESTCD='MAXSEV';
        FATEST='Maximum Severity';
        FASTRESU=' ';
    end;

    if fastat^='NOT DONE' then do;
        If 0 <=AVAL < 2.5 then Sevgrade='NONE';
        Else If 2.5 <=AVAL <=5 then Sevgrade='MILD';
        Else If 5 < AVAL <=10 then Sevgrade='MODERATE';
        Else If aval >10 then Sevgrade='SEVERE';
        avalc=sevgrade;
        aval=input(avalc, ?? sevgrade.);
    end;
    if fastat='NOT DONE' then do;
        aval=.;
        avalc="";
    end;
    if missing(aval) then avalc="";
run;

/*Do not consider the records after unblinding date for MAXSEV parameter*/
proc sql;
    create table temp_ as
    select a.usubjid,a.faobj, a.atptref,fagrp,a.falngrp,
    parcat1,parcat2,studyid,srdom/*,faevintx*/,avisit,avisitn,EXDOSE,ExDOSU,EXSTDTC,EXENDTC,extrt,a.aval,
    'MAXSEV' as FATESTCD,'Maximum Severity' as FATEST,'MAXIMUM' as DTYPE
    from facevd1 a , (select usubjid,faobj,atptref,/*falngrp,*/ max(aval) as AVAL from facevd1 where fatestcd="SEV"
    and CUTUNBFL ne "Y"
        group by usubjid,faobj,atptref/*,falngrp*/) b
    where
        a.usubjid=b.usubjid and
        a.faobj=b.faobj and
        a.atptref=b.atptref and
        a.aval=b.aval and a.CUTUNBFL ne "Y";
quit;

```

```

proc sort data=temp_ ;
    by usubjid faobj atptref parcat1;
run;
proc sort data=temp_ nodupkey;
    by usubjid faobj atptref;
run;

data temp_ ;
    length avalc $200.;
    set temp_ ;
    FASTINT="P0H";
    FAENINT="P7D";
    ADY=.;
    if aval=0 then avalc='NONE';
    if aval=1 then avalc='MILD';
    if aval=2 then avalc='MODERATE';
    if aval=3 then avalc='SEVERE';
run;

proc sort data=_facevd1;
    by USUBJID faobj atptref;
run;
data _facevd1;
    set _facevd1 temp_ ;
run;

*****;
**Reset the value of FATESTCD="G4CRIMET" to maximum severity  *;
*****;

data _facev0 _facev01;
    set _facevd;
    if upcase(FATESTCD)='G4CRIMET' then output _facev0;
    else output _facev01;
run;

data _facevd3;
    set _facevd;
    if upcase(faobj) not in ('REDNESS', 'SWELLING') and FATESTCD='SEV' and upcase(parcat2) ne "SYSTEMIC";
    if 0 < ATPTN <=7 or (missing(ATPTN) and upcase(PARCAT1)='REACTOGENICITY - UNPLANNED
ASSESSMENT');
    if missing(avalc) then aval=.;
    if upcase(avalc)='NONE' then aval=0;
    if upcase(avalc)='MILD' then aval=1;
    if upcase(avalc)='MODERATE' then aval=2;
    if upcase(avalc)='SEVERE' then aval=3;
    if upcase(avalc) in ('POTENTIALLY LIFE THREATENING' 'GRADE 4') then aval=4;
run;

*****;
*Specification 4.2 Derive MAXSEV parameter for injecton site pain *;
*****;

proc sql;
    create table temp3 as select a.usubjid, a.faobj, a.atptref, fagrpid, a.falngrp, parcat1, parcat2, studyid, srctdom, avisit,

```

FDA-CBER-2022-5812-0735967

```

avisitn,
    EXDOSE, ExDOSU, EXSTDTC, EXENDTC, extrt, a.aval, 'MAXSEV' as FATESTCD, 'Maximum Severity' as
FATEST, 'MAXIMUM' as DTYPE from _facevd3 a ,
    (select usubjid, faobj, atptref, max(aval) as AVAL from _facevd3 where CUTUNBFL ne "Y" group by usubjid,
faobj, atptref) b
    where a.usubjid=b.usubjid and a.faobj=b.faobj and a.atptref=b.atptref and a.aval=b.aval and a.CUTUNBFL ne "Y";
quit;

proc sort data=temp3;
    by usubjid faobj atptref parcat1;
run;

proc sort data=temp3 nodupkey;
    by usubjid faobj atptref;
run;

data temp3;
    length avalc $200.;
    set temp3;
    FASTINT="P0H";
    FAENINT="P7D";
    ADY=.;

    if aval=0 then avalc='NONE';
    if aval=1 then avalc='MILD';
    if aval=2 then avalc='MODERATE';
    if aval=3 then avalc='SEVERE';
    if aval=4 then avalc='GRADE 4';
run;

proc sort data=_facevd3;
    by USUBJID faobj atptref;
run;

data _facevd3;
    set _facevd3 temp3;
run;

data _facevd4;
    set facevd;
    if upcase(faobj) not in ('REDNESS', 'SWELLING') and upcase(parcat2) ne "SYSTEMIC";
    if fatestcd not in ('SEV');
run;

data _facevd5;
    set facevd;
    if upcase(faobj) in ('REDNESS', 'SWELLING') and fatestcd in ('OCCUR', 'MINDIAM', 'G4CRIMET') and
upcase(parcat2) ne "SYSTEMIC";
run;

data facevd;
    set _facevd _facevd1 _facevd3 _facevd4 _facevd5;
run;

```

```

proc sort data=facevd;
    by fatestd faobj;
run;

proc sort data=lookup;
    by fatestd faobj;
run;

data facevd(rename=( _paramcd=paramcd _paramn=paramn));
    length param $200;
    merge facevd(in=a) lookup;
    by fatestd faobj;
    if a;
    _paramcd=paramcd;
    _paramn=paramn;

    if faeval="INVESTIGATOR" and fatestd="MAXDIAM" then do;
        _paramcd=tranwrd(strip(paramcd), "MDI", "MAD");
        _paramn=paramn+3;
    end;
    drop paramcd paramn;

    if upcase(FATEST)='DIAMETER' or (faeval="INVESTIGATOR" and fatestd in ("MAXDIAM" "MINDIAM"))
then do;
    PARAM=COMPBL(FAOBJ||' '||trim(FATEST)||' '||"cm");
end;
else do;
    PARAM=COMPBL(FAOBJ||' '||trim(FATEST)||' '||trim(FATESTRESU));
end;
run;

*****;
*Specification 4.3 Derive Medications taken Duration PARAMCD="MEDDUR"*;
*****;

proc sort data=facevd out=_startds;
    by USUBJID ATPTREF EXTRT FAOBJ ADT;
    where index(upcase(FAOBJ), 'MEDICATIONS') and avalc='Y' and CUTUNBFL ne "Y" and FATESTCD in
('FEVMEDGI', 'PAIMEDGI', 'MEDTRTGI', 'MEDTFVPN')
    and upcase(FAEVAL) in ("STUDY SUBJECT", "CAREGIVER", "HEALTHCARE PROFESSIONAL");
run;

data _startds(keep=STUDYID USUBJID ATPTREF EXTRT FAOBJ PARCAT1 PARCAT2 _startmed _stopmed
last_atptn _dur _stop_diary_med
    avisit avisitn EXDOSE ExDOSU EXSTDTC EXENDTC);
    set _startds;
    retain _startmed _dur;
    length _dur 8.;
    by USUBJID ATPTREF EXTRT FAOBJ;

    if first.faobj then do;
        _dur=0;
        _startmed=ADT;
        format _startmed date9.;

```



```

    end;
    _dur=_dur + 1;

if last.faobj then do;
    _stop_diary_med=ADT;
    if ATPTN=last_atptn then do;
        _stopmed=.;
        format _stopmed _stop_diary_med date9.;
        output;
    end;
    else if ATPTN < last_atptn and not missing(ATPTN) then do;
        _stopmed=ADT;
        format _stopmed date9.;
        output;
    end;
end;
run;

proc sort data=facevd out=_endds;
    by USUBJID ATPTREF EXTRT FAOBJ AVAL;
    where upcase(faeval)="INVESTIGATOR" and FATESTCD in ('STPDMEDP', 'STPDFEV', 'STPDPAIN',
'STPDMEDT') and FASTAT="";
run;

proc sort data=facevd out=_ongo(keep=USUBJID ATPTREF EXTRT FAOBJ AVALC rename=(avalc=ongo));
    by USUBJID ATPTREF EXTRT FAOBJ AVALC;
    where upcase(faeval)="INVESTIGATOR" and FATESTCD='MEDTFVPN' and AVALC='Y' and CUTUNBFL ne
"Y";
run;

data _endds(keep=STUDYID USUBJID ATPTREF EXTRT FAOBJ PARCAT1 PARCAT2 _stopmed _stopmedc
last_atptn);
    set _endds;
    by USUBJID ATPTREF EXTRT FAOBJ;
    if last.faobj;
    _stopmed=input(avalc, ??is8601da.);
    _stopmedc =avalc;
run;

proc sort data=_endds;
    by USUBJID ATPTREF EXTRT FAOBJ;
run;

proc sort data=_startds;
    by USUBJID ATPTREF EXTRT FAOBJ;
run;

data _strtend;
    merge _startds _endds;
    by USUBJID ATPTREF EXTRT FAOBJ;
run;

data _strtend;
    merge _strtend(in=a) _ongo;

```

```

    by USUBJID ATPTREF EXTRT FAOBJ;
    if a;
run;

```

```

data _strtend;
    set _strtend;
    length PARAM $200 PARAMCD $8;
    PARAM='Medications Duration';
    PARAMCD='MEDDUR';
    PARAMN=195;
    AVAL=_stopmed-_startmed + 1;

    if missing(AVAL) then AVALC='MISSING';
    else AVALC=";
    output;
run;

```

```

data facevd;
    set facevd _strtend;
run;

```

```

*****;
**Specification 4.4 Derive analysis flags knowvdf1/eventdf1/knwovfl/eventfl*;
* KNOWVDFL - Day known value flag                *;
* EVENTDFL - Day event value flag                  *;
* KNOWVFL - Known value flag excluding e-diary error      *;
* EVENTFL - Event value flag excluding e-diary error      *;
*****;

```

```

proc sql;
    create table given_dayn as select distinct usubjid, faobj, extrt, atptref , atpt, count(avalc) as cnt from facevd
        group by usubjid, faobj, extrt, atptref , atpt;
quit;

```

```

proc sort data=facevd;
    by usubjid faobj extrt atptref atpt;
run;

```

```

data facevd(drop=cnt);
    merge facevd(in=a) given_dayn;
    by usubjid faobj extrt atptref atpt;

```

```

    if upcase(CLTyp)='DIARY CARD' or (upcase(FAEVAL)='INVESTIGATOR' and fatestd not in ('MEDTFVPN'
'STPDMEDP' 'REL') and ADT ne .) then do;
        knowvdf1='N';
        if cnt > 0 and FASTAT^='NOT DONE' then knowvdf1='Y';
    end;
run;

```

```

proc sql;
    create table day_eventn as select distinct usubjid, faobj, extrt, atptref, atpt, count(avalc) as cnt from facevd
        where index(upcase(PARCAT1), 'REACTOGENICITY') and ((avalc='Y' and upcase(faobj) not in ('REDNESS',
'SWELLING'))

```

```
    or upcase(avalc) in ('MILD', 'MODERATE', 'SEVERE') or (index(upcase(FATESTCD),'DIAM') and ATPT ne "
and aval>=2.5))
```

```
    group by usubjid, faobj, extrt, atptref, atpt order by usubjid, faobj, extrt, atptref, atpt;
quit;
```

```
proc sort data=facevd;
    by usubjid faobj extrt atptref atpt;
run;
```

```
data facevd;
    merge facevd(in=a) day_eventn;
    by usubjid faobj extrt atptref atpt;
```

```
    if upcase(CLTYP)='DIARY CARD' or (upcase(FAEVAL)='INVESTIGATOR' and fatestd not in ('MEDTFVPN'
'STPDMEDP' 'REL') and ADT ne .) then do;
        eventdf1='N';
        if cnt > 0 and FASTAT^='NOT DONE' then eventdf1='Y';
    end;
run;
```

```
proc sql;
    create table bign1 as select distinct usubjid, faobj, extrt, atptref, count(1) as cnt from facevd
        where (upcase(AVALC) in ('Y', 'N') or ^missing(AVALC) or ^missing(aval)) and ^(faobj="MEDICATIONS" and
upcase(FAEVAL) ne "STUDY SUBJECT")
        group by usubjid, faobj, extrt, atptref;
quit;
```

```
proc sort data=facevd(drop=cnt);
    by usubjid faobj extrt atptref;
run;
```

```
data facevd1 facevd2;
    merge facevd(in=a) bign1(in=b);
    by usubjid faobj extrt atptref;
    knowvfl='N';
```

```
    if a and b then do;
        if cnt > 0 then knowvfl='Y';
        output facevd1;
    end;
```

```
    if a and not b then do;
        output facevd2;
    end;
run;
```

```
data facevd;
    set facevd1 facevd2;
run;
```

```
proc sql;
    create table anysevflg as select distinct USUBJID, FAOBJ, EXTRT, ATPTREF, COUNT(1) AS CNT from facevd
        where upcase(parcat2) ^='VACCINATION' and ((avalc='Y' and upcase(faobj) not in ('REDNESS', 'SWELLING'))
or
```

```

        upcase(avalc) in ('MILD', 'MODERATE', 'SEVERE') or (index(upcase(FATESTCD), 'DIAM') and ATPT ne " and
aval>=2.5)) and ^(fatestcd in ("MEDTFVPN" "REL") and faeval="INVESTIGATOR")
        group by USUBJID, FAOBJ, EXTRT, ATPTREF;
quit;

proc sort data=facevd;
    by USUBJID FAOBJ EXTRT ATPTREF;
run;

data facevd1 facevd2;
    merge facevd(in=a) anysevf1g(in=b);
    by USUBJID FAOBJ EXTRT ATPTREF;

    if a and b then do;
        if cnt > 0 then EVENTFL='Y';
        output facevd1;
    end;

    if a and not b then do;
        EVENTFL='N';
        output facevd2;
    end;
run;

data facevd(rename=(faseq=srcseq));
    set facevd1 facevd2;
    length avalcat1 avalcat $100.;

    if FATESTCD in ("DIAMETER" "MAXDIAM") and aval > 0 then do;
        if fastat^='NOT DONE' then do;
            If 0 <= AVAL <= 2.0 then AVALCAT='>0-2.0';
            Else If 2.0 < AVAL <= 5 then AVALCAT='>2.0-5.0';
            Else If 5.0 < AVAL <= 10 then AVALCAT='>5.0-10.0';
            Else If aval >= 10.5 then AVALCAT='>10.0';
            avalcat1=avalcat;
            avalcat1n=input(avalcat1, ?? avalcat.);
        end;
    end;
run;

data facevd_loc;
    set facevd;
run;

*****
*Specification 5 Systemic Events
*****

data facevd;
    set facevd_org;
    if upcase(parcat2)="SYSTEMIC";
run;

data _facevd2;

```

```

set facevd;
if upcase(faobj) in ('FEVER');
if 0 < ATPTN <=7 or (missing(ATPTN) and upcase(PARCAT1)='REACTOGENICITY - UNPLANNED
ASSESSMENT');
run;

```

```

*****;
*Specification 5.1 Derive MAXTEMP parameter for Fever *;
*****;

```

```

proc sql;
  create table temp2 as select distinct a.usubjid, a.faobj, a.atptref, fagrpid, a.falnkgp, parcat1, parcat2, studyid, srcdom,
  a.visit, a.visitn,
  EXDOSE, ExDOSU, EXSTDTC, EXENDTC, extrt, a.vsstresn as aval, a.ftemcatn, 'MAXTEMP' as FATESTCD ,
'MAXIMUM' as DTYPE from _facevd2 a ,
  (select usubjid, faobj, atptref, falnkgp, max(vsstresn) as AVAL, max(ftemcatn) as ftemcatn from _facevd2 where
  ^ (vsstresu='C' and vsstresn>42) and CUTUNBFL ne "Y" group by usubjid, faobj, atptref, falnkgp) b
  where a.usubjid=b.usubjid and a.faobj=b.faobj and a.atptref=b.atptref and a.falnkgp=b.falnkgp and
  a.vsstresn=b.aval and a.CUTUNBFL ne "Y";
quit;

```

```

proc sort data=temp2;
  by usubjid faobj atptref parcat1;
run;

```

```

proc sort data=temp2 nodupkey;
  by usubjid faobj atptref;
run;

```

```

data temp2;
  length ftemcat $200.;
  set temp2;
  where aval ne .;
  FASTINT="P0H";
  FAENINT="P7D";
  ADY=.;
  FATEST='Maximum Temperature';
  FATESTCD='MAXTEMP';

```

```

  if FTEMCATN=. then
    FTEMCAT='Missing';
  FTEMCAT=put(ftemcatn, ftemcat.);
run;

```

```

proc sort data=_facevd2;
  by USUBJID faobj atptref;
run;

```

```

data _facevd2;
  set _facevd2 temp2;
run;

```

```

data _facevd3;
  set facevd;
  if upcase(faobj) not in ('FEVER') and FATESTCD='SEV' and upcase(parcat2) eq "SYSTEMIC";

```

FDA-CBER-2022-5812-0735974

```

    if 0 < ATPTN <=7 or (missing(ATPTN) and upcase(PARCAT1)='REACTOGENICITY - UNPLANNED
ASSESSMENT');
    if missing(avalc) then aval=.;
    if upcase(avalc)='NONE' then aval=0;
    if upcase(avalc)='MILD' then aval=1;
    if upcase(avalc)='MODERATE' then aval=2;
    if upcase(avalc)='SEVERE' then aval=3;
    if upcase(avalc) in ('POTENTIALLY LIFE THREATENING' 'GRADE 4') then aval=4;
run;

*****;
*Specification 5.2 Derive MAXSEV parameter *;
*****;

proc sql;
    create table temp3 as select a.usubjid, a.faobj, a.atptref, fagrp, a.falngrp, parcat1, parcat2, studyid, srcdom, avisit,
    avisitn,
        EXDOSE, ExDOSU, EXSTDTC, EXENDTC, extrt, a.aval, 'MAXSEV' as FATESTCD, 'Maximum Severity' as
FATEST, 'MAXIMUM' as DTYPE from _facevd3 a ,
        (select usubjid, faobj, atptref, falngrp, max(aval) as AVAL from _facevd3 where CUTUNBFL ne "Y" group by
usubjid, faobj, atptref, falngrp) b
        where a.usubjid=b.usubjid and a.faobj=b.faobj and a.atptref=b.atptref and a.falngrp=b.falngrp and a.aval=b.aval
and a.CUTUNBFL ne "Y";
quit;

proc sort data=temp3;
    by usubjid faobj atptref parcat1;
run;

proc sort data=temp3 nodupkey;
    by usubjid faobj atptref;
run;

data temp3;
    length avalc $200.;
    set temp3;
    FASTINT="P0H";
    FAENINT="P7D";
    ADY=.;
    if aval=0 then avalc='NONE';
    if aval=1 then avalc='MILD';
    if aval=2 then avalc='MODERATE';
    if aval=3 then avalc='SEVERE';
    if aval=4 then avalc='GRADE 4';
run;

proc sort data=_facevd3;
    by USUBJID faobj atptref;
run;

data _facevd3;
    set _facevd3 temp3;
run;

data _facevd4;

```

```

    set facevd;
    if upcase(faobj) not in ('FEVER');
    if fatestd not in ('SEV');
run;

data facevd;
    set _facevd2 _facevd3 _facevd4;
run;

proc sort data=facevd;
    by fatestd faobj;
run;

proc sort data=lookup;
    by fatestd faobj;
run;

data facevd;
    length param $200;
    merge facevd(in=a) lookup;
    by fatestd faobj;
    if a;
run;

*****
**Specification 5.3 Derive analysis flags knowvdf/evntdf/knownvfl/eventfl*;
* KNOWVDFL - Day known value flag                *;
* EVENTDFL - Day event value flag                 *;
* KNOWVFL - Known value flag excluding e-diary error *;
* EVENTFL - Event value flag excluding e-diary error *;
*****

proc sql;
    create table given_dayn as select distinct usubjid, faobj, extrt, atptref, atpt, count(avalc) as cnt from facevd
    group by usubjid, faobj, extrt, atptref, atpt;
quit;

proc sort data=facevd;
    by usubjid faobj extrt atptref atpt;
run;

data facevd(drop=cnt);
    merge facevd(in=a) given_dayn;
    by usubjid faobj extrt atptref atpt;

    if upcase(CLTPY)='DIARY CARD' or (upcase(FAEVAL)='INVESTIGATOR' and fatestd not in ('MEDTFVPN'
'STPDMEDP' 'REL') and ADT ne .) then do;
        knowvdf='N';
        if cnt > 0 and FASTAT^='NOT DONE' then knowvdf='Y';
    end;
run;

proc sql;

```

```

create table day_eventn as select distinct usubjid, faobj, extrt, atptref, atpt, count(avalc) as cnt from facevd
where index(upcase(PARCAT1), 'REACTOGENICITY') and ((avalc='Y' and upcase(faobj) not in ('REDNESS',
'SWELLING')) or upcase(avalc) in ('MILD', 'MODERATE', 'SEVERE') or (index(upcase(FATESTCD), 'DIAM') and
ATPT ne " and aval>=2.5))
group by usubjid, faobj, extrt, atptref, atpt order by usubjid, faobj, extrt, atptref, atpt;
quit;

proc sort data=facevd;
by usubjid faobj extrt atptref atpt;
run;

data facevd;
merge facevd(in=a) day_eventn;
by usubjid faobj extrt atptref atpt;

if upcase(CLTP) = 'DIARY CARD' or (upcase(FAEVAL) = 'INVESTIGATOR' and fatestcd not in ('MEDTFVPN'
'STPDMEDP' 'REL') and ADT ne .) then do;
eventdf1='N';
if cnt > 0 and FASTAT^='NOT DONE' then eventdf1='Y';
end;
run;

proc sql;
create table bign1 as select distinct usubjid, faobj, extrt, atptref, count(1) as cnt from facevd
where (upcase(AVALC) in ('Y', 'N') or ^missing(AVALC) or ^missing(aval)) and ^(faobj="MEDICATIONS" and
upcase(FAEVAL) ne "STUDY SUBJECT")
group by usubjid, faobj, extrt, atptref;
quit;

proc sort data=facevd(drop=cnt);
by usubjid faobj extrt atptref;
run;

data facevd1 facevd2;
merge facevd(in=a) bign1(in=b);
by usubjid faobj extrt atptref;
knowvfl='N';

if a and b then do;
if cnt > 0 then knowvfl='Y';
output facevd1;
end;

if a and not b then do;
output facevd2;
end;
run;

data facevd;
set facevd1 facevd2;
run;

proc sql;
create table anysevl as select distinct USUBJID, FAOBJ, EXTRT, ATPTREF, COUNT(1) AS CNT from facevd

```

FDA-CBER-2022-5812-0735977


```

    where upcase(parcat2) ^= 'VACCINATION' and ((avalc='Y' and upcase(faobj) not in ('REDNESS', 'SWELLING'))
or
    upcase(avalc) in ('MILD', 'MODERATE', 'SEVERE') or (index(upcase(FATESTCD), 'DIAM') and ATPT ne " and
aval>=2.5)) and ^(fatestcd in ("MEDTFVPN" "REL") and faeval="INVESTIGATOR")
    group by USUBJID, FAOBJ, EXTRT, ATPTREF;
quit;

proc sort data=facevd;
    by USUBJID FAOBJ EXTRT ATPTREF;
run;

data facevd1 facevd2;
    merge facevd(in=a) anysevflg(in=b);
    by USUBJID FAOBJ EXTRT ATPTREF;

    if a and b then do;
        if cnt > 0 then EVENTFL='Y';
        output facevd1;
    end;

    if a and not b then do;
        EVENTFL='N';
        output facevd2;
    end;
run;

data facevd(rename=(faseq=srcseq));
    set facevd1 facevd2;
run;

data facevd_SYS;
    set facevd;
run;

data facevd;
    set facevd_loc facevd_SYS;
    if upcase(FATESTCD)='REL' then delete;
run;

*****
*Specification 6 Derive EVENTOCC - number of events per USUBJID ATPTREF EXTRT FAOBJ*;
*****
proc sort data=facevd(keep=atptref usubjid faobj fatestcd eventdf1 atptn) out=_biphasic1;
    by atptref usubjid faobj atptn descending eventdf1;
    where upcase(fatestcd)="OCCUR";
run;

proc sort data=_biphasic1 nodupkey;
    by atptref usubjid faobj atptn;
run;

proc sql noprint;
    select max(atptn) into :_maxtptn from _biphasic1;
quit;

```

```

proc transpose data=_biphasic1 out=_biphasic2 prefix=day_ LET;
  id ATPTN;
  by ATPTREF USUBJID FAOBJ;
  var EVENTDFL;
run;

data _biphasic2;
  set _biphasic2;
  length fd eventocc 8 biphasfl event_occured $1;
  label biphasfl="Biphasic Patterns Flag ";
  event_occured="N";
  biphasfl="N";
  eventocc=0;
  array days(7) day_1 day_2 day_3 day_4 day_5 day_6 day_7;

  do i=1 to 7;
    if days(i)="Y" and event_occured="N" then do;
      event_occured="Y";
      fd=i;
      eventocc=1;
    end;

    if i>3 then do;
      if (event_occured="Y" and fd ne i) and days(i)="Y" and days(i-1)in ("N", "") and days(i-2)in ("N", "") then
biphasfl="Y";
    end;

    if i>=3 then do;
      if (event_occured="Y" and fd ne i) and days(i)="Y" and days(i-1) in ("N", "") then
EVENTOCC=EVENTOCC + 1;
    end;
  end;
run;

proc sort data=_biphasic2 (keep=ATPTREF USUBJID FAOBJ BIPHASFL EVENTOCC);
  by ATPTREF USUBJID FAOBJ;
run;

proc sort data=FACEVD;
  by ATPTREF USUBJID FAOBJ;
run;

data FACEVD;
  merge FACEVD _biphasic2;
  by ATPTREF USUBJID FAOBJ;
  PARAM=upcase(substr(PARAM, 1, 1))||lowcase(substr(PARAM, 2, length(PARAM)));
run;

proc sort data=facevd;
  by usubjid parcat1 parcat2 faobj paramcd atpt;
run;

data facevd;

```

```

set facevd;
format adtm datetime.;
adtm=dhms(ad, 0, 0, atm);
run;

proc sql;
  create table temp as select USUBJID, ATPTREF, FAOBJ, EXTRT, min(adtm) as first_adtm, min(ad) as First_adt
  from facevd
  group by USUBJID, ATPTREF, FAOBJ, EXTRT;
quit;

proc sort data=facevd;
  by USUBJID ATPTREF FAOBJ EXTRT;
run;

data facevd;
  merge facevd temp;
  by USUBJID ATPTREF FAOBJ EXTRT;
run;

*****
* Specification 7
* Merge with adsl dataset and keep necessary variables.
* Generate TRTx(N) and derive avisit of unplanned assessment.
* Impute stop date of meds with dose 2 date
*****

%let g_adsl_vars=%str(STUDYID USUBJID SUBJID SITEID AGE AGEU SEX SEXN RACE RACEN ARACE
ARACEN ETHNIC ETHNICN COUNTRY SAFFL ARM ARMCD ACTARM ACTARMCD
TRTSDT TRTSTM TRTSDTM TRTEDT TRTETM TRTEDTM TRT01A TRT02A TRT02AN TRT01AN TRT01P
TRT01PN TRT02P TRT02PN TR01SDT TR01STM TR01SDTM TR01EDT TR01ETM TR01EDTM TR02SDT
TR02STM TR02SDTM TR02EDT TR02ETM TR02EDTM
VAX101DT VAX102DT COHORT COHORTN PHASE PHASEN DOSPLVL DOSPLVLN DOSALVL DOSALVLN
AGEGR1 AGEGR1N AGEGR2 AGEGR2N AGEGR4 AGEGR4N RACEGR1 RACEGR1N VAX101 VAX102
DS30KFL COVBLST MULENRFL HIVFL PEDREAFL UNBLNDDT DS3KFL REACTOFL AGETR01 VAX10U
VAX10UDT RANDFL RAND1FL SAF1FL SAF2FL);

data adsl;
  Set datvprot.adsl;
run;

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

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

data facevd;
  merge _ds1(in=d1) _ds2(in=d2 keep=&g_adsl_vars);
  by USUBJID;
  if d1;

```

```

run;

data facevd;
  set facevd(rename=(avisit=avisit2));
  attrib TRTP length=$100. label="Planned Treatment"
    TRTA length=$100. label="Actual Treatment"
    TRTPN label="Planned Treatment (N)"
    TRTAN label="Actual Treatment (N)" ;

  if atpt eq " or fastat="NOT DONE" then do;knowvdf="";eventdf="";end;

  length avisit $30.;
  avisit=strip(avisit2);
  if avisit eq " and avisitn ne . and int(avisitn) ne avisitn then avisit="UNPLANNED VISIT "||strip(put(avisitn, best.));

  TRTA=TRT01A;
  TRTAN=TRT01AN;
  TRTP=TRT01P;
  TRTPN=TRT01PN;
  if (VAX101='Placebo' and ATPTREF="VACCINATION 1") or (VAX102='Placebo' and
ATPPTREF="VACCINATION 2") then do;
    TRTA='Placebo';TRTAN=9;
  end;
  if (VAX102=' ' and ATPPTREF="VACCINATION 2") then do;
    TRTA=' ';TRTAN=.;
  end;

  if index(parcat1, "- UNPLANNED") then PARCAT1=strip(scan(parcat1, 1, '-'));
  if ATPPTREF="VACCINATION 1" and VAX102DT ne . and paramcd="MEDDUR" and _startmed ne .
    and ATPPTN=last_atptn and (_stopmed>VAX102DT or (_stopmed eq "" and ongo='Y')) then do;
    AVAL=VAX102DT-_startmed + 1;
    AVALC="";
  end;
run;

proc sort data=facevd;
  where atptref ne ";
  by usubjid parcat2 faobj paramcd atptref atpt adt adm;
run;

*****
* Specification 8 Output final data      *;
*****
%let g_adfacevd_vars=%str(SRCDOM SRCSEQ FAGRPID FALNKID FALNKGRP FATEST FATESTCD PARAM
PARAMCD PARAMN FAOBJ PARCAT1 PARCAT2 AVALC AVAL
AVALCAT1 AVALCA1N FASTAT FAREASND FAEVAL AVISITN AVISIT ADT ADTM ADY ATPT ATPTN
ATPTREF FAEVINTX DTYPE FASTINT FAENINT EXDOSE EXTRT EXDOSU
EXSTDTC EXENDTC CLTYP VSORRES VSORRESU VSSTRESN VSSTRESU FTEMCAT FTEMCATN
KNOWVFL EVENTFL KNOWVDFL EVENTDFL CUTUNBFL EVENTOCC TRTA TRTAN TRTP TRTPN);

data datvprot.adfacevd(label='Diary and Non-event Analysis Dataset');
  retain &g_adsl_vars &g_adfacevd_vars;
  set facevd(keep=&g_adsl_vars &g_adfacevd_vars);
  label SRCDOM='Source Data'

```

SRCSEQ='Source Sequence Number'
 PARAMCD='Parameter Code'
 PARAM='Parameter'
 PARAMN='Parameter (N)'
 PARCAT1='Parameter Category 1'
 PARCAT2='Parameter Category 2'
 AVALC='Analysis Value (C)'
 AVAL='Analysis Value'
 AVALCAT1='Analysis Value Category 1'
 AVALCA1N='Analysis Value Category 1 (N)'
 AVISITN='Analysis Visit (N)'
 AVISIT='Analysis Visit'
 ADT='Analysis Date'
 FASTINT='Evaluation Interval Start'
 FAENINT='Evaluation Interval End'
 ADY='Analysis Relative Day'
 ATPT='Analysis Timepoint'
 ATPTREF='Analysis Timepoint Reference'
 CLTYP='Collection Type'
 FTEMCAT='Fever Temperature Category'
 FTEMCATN='Fever Temperature Category (N)'
 KNOWVFL='Known Value Flag'
 EVENTFL='Event Value Flag'
 KNOWVDFL='Day Known Value Flag'
 EVENTDFL='Day Event Value Flag'
 DTYPE='Derivation Type'
 ATPTN='Analysis Timepoint (N)'
 TRTP='Planned Treatment'
 TRTPN='Planned Treatment (N)'
 TRTA='Actual Treatment'
 TRTAN='Actual Treatment (N)'
 EVENTOCC='Occurrences of Event'
 ADTM='Analysis Datetime'
 CUTUNBFL = 'Cut after Unblinding Date Flag';

run;

proc printto;

run;