

```

*****
** Program Name   : adce-s020-se-sev-ped-saf.sas                **
** Date Created  : 11Mar2021                                     **
** Programmer Name : (b) (6)                                     **
** Purpose       : Create adce-s020-se-sev-ped-saf             **
** Input data    : adfacevd adsl                                **
** Output file   : adce-s020-se-sev-ped-saf.html               **
*****

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

**Setup the environment**
%let prot=/Volumes/app/cdars/prod/sites/cdars4/prjC459/nda2_unblinded_esub/euaext_esub_adam/saseng/cdisc3_0;
libname datvprot "&prot./data_vai" access=readonly;

%let codename=adce-s020-se-sev-ped-saf;
%let outlog=&prot./analysis/esub/logs/&codename..log;
%let outtable=&prot./analysis/esub/output/&codename..html;

proc printto log="&outlog" new;
run;
*****
* Specification 1
* Create foramt
*****

Proc format;
  value SEV
    0="Any"
    1="Mild"
    2="Moderate"
    3="Severe"
    4="Grade 4"
  ;
  value VAC
    1="1"
    2="2"
    3="3"
    99="ANY"
  ;
run;

*****
* Specification 2
* Input source data adfacevd and adsl
*****

data g_adsl_dsin;
  set DATVPROT.ADSL;
  where SAFFL eq 'Y' and PEDREAFL="Y" and hivfl ne "Y" and agegr4n ne . and MULENRFL ne "Y";
run;

```

```
data g_a_dsin;
  set DATVPROT.adfacevd;
  where SAFFL eq 'Y' and CUTUNBFL ne "Y" and PEDREAFL="Y" and hivfl ne "Y" and knowvfl="Y" and
  agegr4n ne . and MULENRFL ne "Y";
```

```
  if TRTAN in (8) then
    do;
      newtrtn=1;
      newtrt=coalescec("BNT162b2 (30 (*ESC*){unicode 03BC}g)~{line}", TRTA);
      output;
    end;
```

```
  if TRTAN in (9) then
    do;
      newtrtn=2;
      newtrt=coalescec("Placebo~{line}", TRTA);
      output;
    end;
```

```
run;
```

```
data g_a_dsin;
  set g_a_dsin;
  atptrefn=input(compress(atptref, ", 'A'), ??best.);
  atptref=compress(atptref, ", 'A');
  output;
```

```
  if atptref ne "";
  atptref="Any dose";
  atptrefn=99;
  output;
```

```
run;
```

```
*****;
**Regarding medication errors, subset for Reactogenicity analysis          **;
**1.Count subjects in what they received at Dose 1 for post Dose 1 summary.  **;
**2.Remove subjects from post Dose 2 summary.                             **;
**3.Count subjects in active for after any dose summary.                   **;
*****;
```

```
data g_a_dsin;
  set g_a_dsin;
  where trta ne ";
  if VAX101 ne VAX102 and cmiss(VAX101,VAX102)=0 then do;
    if atptrefn=2 then delete;
    if atptrefn=99 then do;
      TRTAN=TRT01AN;TRTA=TRT01A;
      if TRTAN=8 then do; newtrtn = 1; newtrt = "BNT162b2 (30 (*ESC*){unicode 03BC}g)~{line}"; end;
      if TRTAN=9 then do; newtrtn = 2; newtrt = "Placebo~{line}"; end;
    end;
  end;
```

```
*****;
* Specification 3 *
```

```

* 1) Select all necessary parameters
* 2) Create flags for Any Local Reaction and Any Dose rows
* 3) Create order variables for next statistic analyses
* 4) Merge adsl and analysis dataset
*****

```

```

proc sql;
  create table a1 as select distinct newtrt, usubjid, faobj , atptref from
    g_a_dsin where upcase(FATESTCD)='OCCUR';
quit;

```

```

proc sql;
  create table a2 as select distinct newtrt, usubjid, faobj , atptref from a1
    except select distinct newtrt, usubjid , faobj , atptref from g_a_dsin where
      FATESTCD='MAXSEV';
quit;

```

```

proc sort data=a2;
  by newtrt usubjid faobj atptref;
quit;

```

```

proc sort data=g_a_dsin out=facevd;
  by newtrt usubjid faobj atptref;
quit;

```

```

data a3;
  merge facevd(in=a) a2(in=b);
  by newtrt usubjid faobj atptref;

  if fatestd='OCCUR';

  if a and b then
    output;

```

```

run;

proc sort data=a3;
  by faobj newtrt usubjid atptref ady newtrtn;
quit;

```

```

data a4(drop=paramcd);
  set a3;
  by faobj newtrt usubjid atptref ady newtrtn;

  if first.atptref;
  aval=0;
  avalc='NONE';
  knowvfl='Y';
  fatestd='MAXSEV';
run;

```

```

data _param;
  set g_a_dsin;

  if fatestd='MAXSEV';

```

```

    keep faobj paramcd;
run;

proc sort nodupkey data=_param;
    by faobj paramcd;
run;

data a4;
    merge a4(in=a) _param;
    by faobj;

    if a;
run;

data g_a_dsin;
    set g_a_dsin a4;
run;

data _a_dsin;
    set g_a_dsin;
    length _faobj_order 8 _faobj_label $100.;

    if missing(aval) then
        aval=0;

    if missing(avalc) then
        avalc="NONE";

    if upcase(paramcd)="MAXTEMP" then
        do;
            _faobj_order=1/100;

            if upcase(paramcd) in ('MAXTEMP') then
                do;
                    _faobj_order=_faobj_order + 1;
                    _faobj_label=trim(propcase(FAOBJ));
                end;
            else if upcase(paramcd) in ('MAXSFAT', 'MAXCHIL', 'MAXDEAP', 'MASIRR',
                'MASINSL', '') then
                do;
                    _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
                    _faobj_order=_faobj_order + 2;
                end;
            else if upcase(paramcd) in ('MAXSHEA') then
                do;
                    _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
                    _faobj_order=_faobj_order + 2;
                end;
            else if upcase(paramcd) in ('MAXSNV', 'MAXSVOM', 'MAXSNU') then
                do;
                    _faobj_order=_faobj_order + 3;
                    _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super e}";
                end;
            else if upcase(paramcd) in ('MAXDIAR') then

```

```

do;
    _faobj_order=_faobj_order + 4;
    _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super f}";
end;
else if upcase(paramcd) in ('FEVMEDGI', 'PAIMEDGI', 'MEDTFVPN') then
do;
    _faobj_order=_faobj_order + 99;
    _faobj_label="Use of antipyretic or pain medication"||"(*ESC*){super h}";
end;
else if upcase(paramcd) in ('OCRASH', 'OCDECAP', 'OCHIVES', 'OCDECSL') then
do;
    _faobj_order=_faobj_order + 5;
    _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super g}";
end;
else if upcase(paramcd) in ('MASNMP', 'MAAMP', 'MAXSMP') then
do;
    _faobj_label=trim("New or worsened muscle pain")||"(*ESC*){super d}";
    _faobj_order=_faobj_order + 6;
end;
else if upcase(paramcd) in ('MANJP', 'MAXAJP', 'MAXSJP') then
do;
    _faobj_label=trim("New or worsened joint pain")||"(*ESC*){super d}";
    _faobj_order=_faobj_order + 7;
end;
else
do;
    _faobj_order=_faobj_order + 8;
    _faobj_label=trim(propcase(FAOBJ));
end;
output;
end;

if upcase(paramcd)="MSEVNAUS" then
do;
    _faobj_order=2/100;

    if upcase(paramcd) in ('MAXTEMP') then
do;
    _faobj_order=_faobj_order + 1;
    _faobj_label=trim(propcase(FAOBJ));
end;
else if upcase(paramcd) in ('MAXSFAT', 'MAXCHIL', 'MAXDEAP', 'MASIRR',
'MASINSL', '') then
do;
    _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
    _faobj_order=_faobj_order + 2;
end;
else if upcase(paramcd) in ('MAXSHEA') then
do;
    _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
    _faobj_order=_faobj_order + 2;
end;
else if upcase(paramcd) in ('MAXSNV', 'MAXSVOM', 'MAXSNU') then
do;

```

```

        _faobj_order=_faobj_order + 3;
        _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super e}";
    end;
else if upcase(paramcd) in ('MAXDIAR') then
    do;
        _faobj_order=_faobj_order + 4;
        _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super f}";
    end;
else if upcase(paramcd) in ('FEVMEDGI', 'PAIMEDGI', 'MEDTFVPN') then
    do;
        _faobj_order=_faobj_order + 99;
        _faobj_label="Use of antipyretic or pain medication"||"(*ESC*){super h}";
    end;
else if upcase(paramcd) in ('OCRASH', 'OCDECAP', 'OCHIVES', 'OCDECSL') then
    do;
        _faobj_order=_faobj_order + 5;
        _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super g}";
    end;
else if upcase(paramcd) in ('MASNMP', 'MAAMP', 'MAXSMP') then
    do;
        _faobj_label=trim("New or worsened muscle pain")||"(*ESC*){super d}";
        _faobj_order=_faobj_order + 6;
    end;
else if upcase(paramcd) in ('MANJP', 'MAXAJP', 'MAXSJP') then
    do;
        _faobj_label=trim("New or worsened joint pain")||"(*ESC*){super d}";
        _faobj_order=_faobj_order + 7;
    end;
else
    do;
        _faobj_order=_faobj_order + 8;
        _faobj_label=trim(propcase(FAOBJ));
    end;
output;
end;

if upcase(paramcd)="MAXSFAT" then
    do;
        _faobj_order=3/100;

        if upcase(paramcd) in ('MAXTEMP') then
            do;
                _faobj_order=_faobj_order + 1;
                _faobj_label=trim(propcase(FAOBJ));
            end;
        else if upcase(paramcd) in ('MAXSFAT', 'MAXCHIL', 'MAXDEAP', 'MASIRR',
            'MASINSL', '') then
            do;
                _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
                _faobj_order=_faobj_order + 2;
            end;
        else if upcase(paramcd) in ('MAXSHEA') then
            do;
                _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
            end;
        end;
    end;
end;

```

FDA-CBER-2022-5812-0735669

```

        _faobj_order=_faobj_order + 2;
    end;
else if upcase(paramcd) in ('MAXSNV', 'MAXSVOM' 'MAXSNU') then
    do;
        _faobj_order=_faobj_order + 3;
        _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super e}";
    end;
else if upcase(paramcd) in ('MAXDIAR') then
    do;
        _faobj_order=_faobj_order + 4;
        _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super f}";
    end;
else if upcase(paramcd) in ('FEVMEDGI', 'PAIMEDGI', 'MEDTFVPN') then
    do;
        _faobj_order=_faobj_order + 99;
        _faobj_label="Use of antipyretic or pain medication"||"(*ESC*){super h}";
    end;
else if upcase(paramcd) in ('OCRASH', 'OCDECAP', 'OCHIVES', 'OCDECSL') then
    do;
        _faobj_order=_faobj_order + 5;
        _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super g}";
    end;
else if upcase(paramcd) in ('MASNMP', 'MAAMP', 'MAXSMP') then
    do;
        _faobj_label=trim("New or worsened muscle pain")||"(*ESC*){super d}";
        _faobj_order=_faobj_order + 6;
    end;
else if upcase(paramcd) in ('MANJP', 'MAXAJP', 'MAXSJP') then
    do;
        _faobj_label=trim("New or worsened joint pain")||"(*ESC*){super d}";
        _faobj_order=_faobj_order + 7;
    end;
else
    do;
        _faobj_order=_faobj_order + 8;
        _faobj_label=trim(propcase(FAOBJ));
    end;
output;
end;

if upcase(paramcd)="MAXSHEA" then
    do;
        _faobj_order=4/100;

        if upcase(paramcd) in ('MAXTEMP') then
            do;
                _faobj_order=_faobj_order + 1;
                _faobj_label=trim(propcase(FAOBJ));
            end;
        else if upcase(paramcd) in ('MAXSFAT', 'MAXCHIL', 'MAXDEAP', 'MASIRR',
            'MASINSL', '') then
            do;
                _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
                _faobj_order=_faobj_order + 2;
            end;
        end;
    end;
end;

```

FDA-CBER-2022-5812-0735670

```

        end;
    else if upcase(paramcd) in ('MAXSHEA') then
        do;
            _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
            _faobj_order=_faobj_order + 2;
        end;
    else if upcase(paramcd) in ('MAXSNV', 'MAXSVOM' 'MAXSNU') then
        do;
            _faobj_order=_faobj_order + 3;
            _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super e}";
        end;
    else if upcase(paramcd) in ('MAXDIAR') then
        do;
            _faobj_order=_faobj_order + 4;
            _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super f}";
        end;
    else if upcase(paramcd) in ('FEVMEDGI', 'PAIMEDGI', 'MEDTFVPN') then
        do;
            _faobj_order=_faobj_order + 99;
            _faobj_label="Use of antipyretic or pain medication"||"(*ESC*){super h}";
        end;
    else if upcase(paramcd) in ('OCRASH', 'OCDECAP', 'OCHIVES', 'OCDECSL') then
        do;
            _faobj_order=_faobj_order + 5;
            _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super g}";
        end;
    else if upcase(paramcd) in ('MASNMP', 'MAAMP', 'MAXSMP') then
        do;
            _faobj_label=trim("New or worsened muscle pain")||"(*ESC*){super d}";
            _faobj_order=_faobj_order + 6;
        end;
    else if upcase(paramcd) in ('MANJP', 'MAXAJP', 'MAXSJP') then
        do;
            _faobj_label=trim("New or worsened joint pain")||"(*ESC*){super d}";
            _faobj_order=_faobj_order + 7;
        end;
    else
        do;
            _faobj_order=_faobj_order + 8;
            _faobj_label=trim(propcase(FAOBJ));
        end;
    output;
end;

if upcase(paramcd)="MAXCHIL" then
do;
    _faobj_order=5/100;

    if upcase(paramcd) in ('MAXTEMP') then
        do;
            _faobj_order=_faobj_order + 1;
            _faobj_label=trim(propcase(FAOBJ));
        end;
    else if upcase(paramcd) in ('MAXSFAT', 'MAXCHIL', 'MAXDEAP', 'MASIRR',

```

FDA-CBER-2022-5812-0735671



```

'MASINSL', ") then
    do;
        _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
        _faobj_order=_faobj_order + 2;
    end;
else if upcase(paramcd) in ('MAXSHEA') then
    do;
        _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
        _faobj_order=_faobj_order + 2;
    end;
else if upcase(paramcd) in ('MAXSNV', 'MAXSVOM' 'MAXSNU') then
    do;
        _faobj_order=_faobj_order + 3;
        _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super e}";
    end;
else if upcase(paramcd) in ('MAXDIAR') then
    do;
        _faobj_order=_faobj_order + 4;
        _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super f}";
    end;
else if upcase(paramcd) in ('FEVMEDGI', 'PAIMEDGI', 'MEDTFVPN') then
    do;
        _faobj_order=_faobj_order + 99;
        _faobj_label="Use of antipyretic or pain medication"||"(*ESC*){super h}";
    end;
else if upcase(paramcd) in ('OCRASH', 'OCDECAP', 'OCHIVES', 'OCDECSL') then
    do;
        _faobj_order=_faobj_order + 5;
        _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super g}";
    end;
else if upcase(paramcd) in ('MASNMP', 'MAAMP', 'MAXSMP') then
    do;
        _faobj_label=trim("New or worsened muscle pain")||"(*ESC*){super d}";
        _faobj_order=_faobj_order + 6;
    end;
else if upcase(paramcd) in ('MANJP', 'MAXAJP', 'MAXSJP') then
    do;
        _faobj_label=trim("New or worsened joint pain")||"(*ESC*){super d}";
        _faobj_order=_faobj_order + 7;
    end;
else
    do;
        _faobj_order=_faobj_order + 8;
        _faobj_label=trim(propcase(FAOBJ));
    end;
output;
end;

if upcase(paramcd)="MAXSVOM" then
    do;
        _faobj_order=6/100;

        if upcase(paramcd) in ('MAXTEMP') then
            do;

```

```

        _faobj_order=_faobj_order + 1;
        _faobj_label=trim(propcase(FAOBJ));
    end;
else if upcase(paramcd) in ('MAXSFAT', 'MAXCHIL', 'MAXDEAP', 'MASIRR',
    'MASINSL', '') then
    do;
        _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
        _faobj_order=_faobj_order + 2;
    end;
else if upcase(paramcd) in ('MAXSHEA') then
    do;
        _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
        _faobj_order=_faobj_order + 2;
    end;
else if upcase(paramcd) in ('MAXSNV', 'MAXSVOM', 'MAXSNU') then
    do;
        _faobj_order=_faobj_order + 3;
        _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super e}";
    end;
else if upcase(paramcd) in ('MAXDIAR') then
    do;
        _faobj_order=_faobj_order + 4;
        _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super f}";
    end;
else if upcase(paramcd) in ('FEVMEDGI', 'PAIMEDGI', 'MEDTFVPN') then
    do;
        _faobj_order=_faobj_order + 99;
        _faobj_label="Use of antipyretic or pain medication"||"(*ESC*){super h}";
    end;
else if upcase(paramcd) in ('OCRASH', 'OCDECAP', 'OCHIVES', 'OCDECSL') then
    do;
        _faobj_order=_faobj_order + 5;
        _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super g}";
    end;
else if upcase(paramcd) in ('MASNMP', 'MAAMP', 'MAXSMP') then
    do;
        _faobj_label=trim("New or worsened muscle pain")||"(*ESC*){super d}";
        _faobj_order=_faobj_order + 6;
    end;
else if upcase(paramcd) in ('MANJP', 'MAXAJP', 'MAXSJP') then
    do;
        _faobj_label=trim("New or worsened joint pain")||"(*ESC*){super d}";
        _faobj_order=_faobj_order + 7;
    end;
else
    do;
        _faobj_order=_faobj_order + 8;
        _faobj_label=trim(propcase(FAOBJ));
    end;
output;
end;

if upcase(paramcd)="MAXDIAR" then
    do;

```

```

_faobj_order=7/100;

if upcase(paramcd) in ('MAXTEMP') then
  do;
    _faobj_order=_faobj_order + 1;
    _faobj_label=trim(propcase(FAOBJ));
  end;
else if upcase(paramcd) in ('MAXSFAT', 'MAXCHIL', 'MAXDEAP', 'MASIRR',
  'MASINSL', '') then
  do;
    _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
    _faobj_order=_faobj_order + 2;
  end;
else if upcase(paramcd) in ('MAXSHEA') then
  do;
    _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
    _faobj_order=_faobj_order + 2;
  end;
else if upcase(paramcd) in ('MAXSNV', 'MAXSVOM', 'MAXSNU') then
  do;
    _faobj_order=_faobj_order + 3;
    _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super e}";
  end;
else if upcase(paramcd) in ('MAXDIAR') then
  do;
    _faobj_order=_faobj_order + 4;
    _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super f}";
  end;
else if upcase(paramcd) in ('FEVMEDGI', 'PAIMEDGI', 'MEDTFVPN') then
  do;
    _faobj_order=_faobj_order + 99;
    _faobj_label="Use of antipyretic or pain medication"||"(*ESC*){super h}";
  end;
else if upcase(paramcd) in ('OCRASH', 'OCDECAP', 'OCHIVES', 'OCDECSL') then
  do;
    _faobj_order=_faobj_order + 5;
    _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super g}";
  end;
else if upcase(paramcd) in ('MASNMP', 'MAAMP', 'MAXSMP') then
  do;
    _faobj_label=trim("New or worsened muscle pain")||"(*ESC*){super d}";
    _faobj_order=_faobj_order + 6;
  end;
else if upcase(paramcd) in ('MANJP', 'MAXAJP', 'MAXSJP') then
  do;
    _faobj_label=trim("New or worsened joint pain")||"(*ESC*){super d}";
    _faobj_order=_faobj_order + 7;
  end;
else
  do;
    _faobj_order=_faobj_order + 8;
    _faobj_label=trim(propcase(FAOBJ));
  end;
output;

```

```

end;

if upcase(paramcd)="FEVMEDGI" then
do;
    _faobj_order=8/100;

    if upcase(paramcd) in ('MAXTEMP') then
        do;
            _faobj_order=_faobj_order + 1;
            _faobj_label=trim(propcase(FAOBJ));
        end;
    else if upcase(paramcd) in ('MAXSFAT', 'MAXCHIL', 'MAXDEAP', 'MASIRR',
        'MASINSL', ") then
        do;
            _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
            _faobj_order=_faobj_order + 2;
        end;
    else if upcase(paramcd) in ('MAXSHEA') then
        do;
            _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
            _faobj_order=_faobj_order + 2;
        end;
    else if upcase(paramcd) in ('MAXSNV', 'MAXSVOM' 'MAXSNU') then
        do;
            _faobj_order=_faobj_order + 3;
            _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super e}";
        end;
    else if upcase(paramcd) in ('MAXDIAR') then
        do;
            _faobj_order=_faobj_order + 4;
            _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super f}";
        end;
    else if upcase(paramcd) in ('FEVMEDGI', 'PAIMEDGI', 'MEDTFVPN') then
        do;
            _faobj_order=_faobj_order + 99;
            _faobj_label="Use of antipyretic or pain medication"||"(*ESC*){super h}";
        end;
    else if upcase(paramcd) in ('OCRASH', 'OCDECAP', 'OCHIVES', 'OCDECSL') then
        do;
            _faobj_order=_faobj_order + 5;
            _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super g}";
        end;
    else if upcase(paramcd) in ('MASNMP', 'MAAMP', 'MAXSMP') then
        do;
            _faobj_label=trim("New or worsened muscle pain")||"(*ESC*){super d}";
            _faobj_order=_faobj_order + 6;
        end;
    else if upcase(paramcd) in ('MANJP', 'MAXAJP', 'MAXSJP') then
        do;
            _faobj_label=trim("New or worsened joint pain")||"(*ESC*){super d}";
            _faobj_order=_faobj_order + 7;
        end;
    else
        do;

```

```

        _faobj_order=_faobj_order + 8;
        _faobj_label=trim(propcase(FAOBJ));
    end;
output;
end;

if upcase(paramcd)="PAIMEDGI" then
do;
    _faobj_order=9/100;

    if upcase(paramcd) in ('MAXTEMP') then
        do;
            _faobj_order=_faobj_order + 1;
            _faobj_label=trim(propcase(FAOBJ));
        end;
    else if upcase(paramcd) in ('MAXSFAT', 'MAXCHIL', 'MAXDEAP', 'MASIRR',
        'MASINSL', '') then
        do;
            _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
            _faobj_order=_faobj_order + 2;
        end;
    else if upcase(paramcd) in ('MAXSHEA') then
        do;
            _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
            _faobj_order=_faobj_order + 2;
        end;
    else if upcase(paramcd) in ('MAXSNV', 'MAXSVOM', 'MAXSNU') then
        do;
            _faobj_order=_faobj_order + 3;
            _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super e}";
        end;
    else if upcase(paramcd) in ('MAXDIAR') then
        do;
            _faobj_order=_faobj_order + 4;
            _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super f}";
        end;
    else if upcase(paramcd) in ('FEVMEDGI', 'PAIMEDGI', 'MEDTFVPN') then
        do;
            _faobj_order=_faobj_order + 99;
            _faobj_label="Use of antipyretic or pain medication"||"(*ESC*){super h}";
        end;
    else if upcase(paramcd) in ('OCRASH', 'OCDECAP', 'OCHIVES', 'OCDECSL') then
        do;
            _faobj_order=_faobj_order + 5;
            _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super g}";
        end;
    else if upcase(paramcd) in ('MASNMP', 'MAAMP', 'MAXSMP') then
        do;
            _faobj_label=trim("New or worsened muscle pain")||"(*ESC*){super d}";
            _faobj_order=_faobj_order + 6;
        end;
    else if upcase(paramcd) in ('MANJP', 'MAXAJP', 'MAXSJP') then
        do;
            _faobj_label=trim("New or worsened joint pain")||"(*ESC*){super d}";

```

FDA-CBER-2022-5812-0735676

```

        _faobj_order=_faobj_order + 7;
    end;
else
    do;
        _faobj_order=_faobj_order + 8;
        _faobj_label=trim(propcase(FAOBJ));
    end;
output;
end;

if upcase(paramcd)="MEDTFVPN" then
    do;
        _faobj_order=10/100;

        if upcase(paramcd) in ('MAXTEMP') then
            do;
                _faobj_order=_faobj_order + 1;
                _faobj_label=trim(propcase(FAOBJ));
            end;
        else if upcase(paramcd) in ('MAXSFAT', 'MAXCHIL', 'MAXDEAP', 'MASIRR',
            'MASINSL', '') then
            do;
                _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
                _faobj_order=_faobj_order + 2;
            end;
        else if upcase(paramcd) in ('MAXSHEA') then
            do;
                _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
                _faobj_order=_faobj_order + 2;
            end;
        else if upcase(paramcd) in ('MAXSNV', 'MAXSVOM', 'MAXSNU') then
            do;
                _faobj_order=_faobj_order + 3;
                _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super e}";
            end;
        else if upcase(paramcd) in ('MAXDIAR') then
            do;
                _faobj_order=_faobj_order + 4;
                _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super f}";
            end;
        else if upcase(paramcd) in ('FEVMEDGI', 'PAIMEDGI', 'MEDTFVPN') then
            do;
                _faobj_order=_faobj_order + 99;
                _faobj_label="Use of antipyretic or pain medication"||"(*ESC*){super h}";
            end;
        else if upcase(paramcd) in ('OCRASH', 'OCDECAP', 'OCHIVES', 'OCDECSL') then
            do;
                _faobj_order=_faobj_order + 5;
                _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super g}";
            end;
        else if upcase(paramcd) in ('MASNMP', 'MAAMP', 'MAXSMP') then
            do;
                _faobj_label=trim("New or worsened muscle pain")||"(*ESC*){super d}";
                _faobj_order=_faobj_order + 6;
            end;
        end;
    end;
end;

```

FDA-CBER-2022-5812-0735677

```

        end;
    else if upcase(paramcd) in ('MANJP', 'MAXAJP', 'MAXSJP') then
        do;
            _faobj_label=trim("New or worsened joint pain")||"(*ESC*){super d}";
            _faobj_order=_faobj_order + 7;
        end;
    else
        do;
            _faobj_order=_faobj_order + 8;
            _faobj_label=trim(propcase(FAOBJ));
        end;
    output;
end;

if upcase(paramcd)="OCHIVES" then
    do;
        _faobj_order=11/100;

        if upcase(paramcd) in ('MAXTEMP') then
            do;
                _faobj_order=_faobj_order + 1;
                _faobj_label=trim(propcase(FAOBJ));
            end;
        else if upcase(paramcd) in ('MAXSFAT', 'MAXCHIL', 'MAXDEAP', 'MASIRR',
            'MASINSL', ") then
            do;
                _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
                _faobj_order=_faobj_order + 2;
            end;
        else if upcase(paramcd) in ('MAXSHEA') then
            do;
                _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
                _faobj_order=_faobj_order + 2;
            end;
        else if upcase(paramcd) in ('MAXSNV', 'MAXSVOM' 'MAXSNU') then
            do;
                _faobj_order=_faobj_order + 3;
                _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super e}";
            end;
        else if upcase(paramcd) in ('MAXDIAR') then
            do;
                _faobj_order=_faobj_order + 4;
                _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super f}";
            end;
        else if upcase(paramcd) in ('FEVMEDGI', 'PAIMEDGI', 'MEDTFVPN') then
            do;
                _faobj_order=_faobj_order + 99;
                _faobj_label="Use of antipyretic or pain medication"||"(*ESC*){super h}";
            end;
        else if upcase(paramcd) in ('OCRASH', 'OCDECAP', 'OCHIVES', 'OCDECSL') then
            do;
                _faobj_order=_faobj_order + 5;
                _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super g}";
            end;
        end;
    end;
end;

```

FDA-CBER-2022-5812-0735678

```

else if upcase(paramcd) in ('MASNMP', 'MAAMP', 'MAXSMP') then
  do;
    _faobj_label=trim("New or worsened muscle pain")||"(*ESC*){super d}";
    _faobj_order=_faobj_order + 6;
  end;
else if upcase(paramcd) in ('MANJP', 'MAXAJP', 'MAXSJP') then
  do;
    _faobj_label=trim("New or worsened joint pain")||"(*ESC*){super d}";
    _faobj_order=_faobj_order + 7;
  end;
else
  do;
    _faobj_order=_faobj_order + 8;
    _faobj_label=trim(propcase(FAOBJ));
  end;
output;
end;

if upcase(paramcd)="OCDECSL" then
  do;
    _faobj_order=12/100;

    if upcase(paramcd) in ('MAXTEMP') then
      do;
        _faobj_order=_faobj_order + 1;
        _faobj_label=trim(propcase(FAOBJ));
      end;
    else if upcase(paramcd) in ('MAXSFAT', 'MAXCHIL', 'MAXDEAP', 'MASIRR',
      'MASINSL', '') then
      do;
        _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
        _faobj_order=_faobj_order + 2;
      end;
    else if upcase(paramcd) in ('MAXSHEA') then
      do;
        _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
        _faobj_order=_faobj_order + 2;
      end;
    else if upcase(paramcd) in ('MAXSNV', 'MAXSVOM', 'MAXSNU') then
      do;
        _faobj_order=_faobj_order + 3;
        _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super e}";
      end;
    else if upcase(paramcd) in ('MAXDIAR') then
      do;
        _faobj_order=_faobj_order + 4;
        _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super f}";
      end;
    else if upcase(paramcd) in ('FEVMEDGI', 'PAIMEDGI', 'MEDTFVPN') then
      do;
        _faobj_order=_faobj_order + 99;
        _faobj_label="Use of antipyretic or pain medication"||"(*ESC*){super h}";
      end;
    else if upcase(paramcd) in ('OCRASH', 'OCDECAP', 'OCHIVES', 'OCDECSL') then

```

FDA-CBER-2022-5812-0735679



```

do;
    _faobj_order=_faobj_order + 5;
    _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super g}";
end;
else if upcase(paramcd) in ('MASNMP', 'MAAMP', 'MAXSMP') then
do;
    _faobj_label=trim("New or worsened muscle pain")||"(*ESC*){super d}";
    _faobj_order=_faobj_order + 6;
end;
else if upcase(paramcd) in ('MANJP', 'MAXAJP', 'MAXSJP') then
do;
    _faobj_label=trim("New or worsened joint pain")||"(*ESC*){super d}";
    _faobj_order=_faobj_order + 7;
end;
else
do;
    _faobj_order=_faobj_order + 8;
    _faobj_label=trim(propcase(FAOBJ));
end;
output;
end;

if upcase(paramcd)="OCRASH" then
do;
    _faobj_order=13/100;

    if upcase(paramcd) in ('MAXTEMP') then
do;
    _faobj_order=_faobj_order + 1;
    _faobj_label=trim(propcase(FAOBJ));
end;
else if upcase(paramcd) in ('MAXSFAT', 'MAXCHIL', 'MAXDEAP', 'MASIRR',
'MASINSL', ") then
do;
    _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
    _faobj_order=_faobj_order + 2;
end;
else if upcase(paramcd) in ('MAXSHEA') then
do;
    _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
    _faobj_order=_faobj_order + 2;
end;
else if upcase(paramcd) in ('MAXSNV', 'MAXSVOM', 'MAXSNU') then
do;
    _faobj_order=_faobj_order + 3;
    _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super e}";
end;
else if upcase(paramcd) in ('MAXDIAR') then
do;
    _faobj_order=_faobj_order + 4;
    _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super f}";
end;
else if upcase(paramcd) in ('FEVMEDGI', 'PAIMEDGI', 'MEDTFVPN') then
do;

```

```

        _faobj_order=_faobj_order + 99;
        _faobj_label="Use of antipyretic or pain medication"||"(*ESC*){super h}";
    end;
else if upcase(paramcd) in ('OCRASH', 'OCDECAP', 'OCHIVES', 'OCDECSL') then
    do;
        _faobj_order=_faobj_order + 5;
        _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super g}";
    end;
else if upcase(paramcd) in ('MASNMP', 'MAAMP', 'MAXSMP') then
    do;
        _faobj_label=trim("New or worsened muscle pain")||"(*ESC*){super d}";
        _faobj_order=_faobj_order + 6;
    end;
else if upcase(paramcd) in ('MANJP', 'MAXAJP', 'MAXSJP') then
    do;
        _faobj_label=trim("New or worsened joint pain")||"(*ESC*){super d}";
        _faobj_order=_faobj_order + 7;
    end;
else
    do;
        _faobj_order=_faobj_order + 8;
        _faobj_label=trim(propcase(FAOBJ));
    end;
output;
end;

if upcase(paramcd)="MASNMP" then
    do;
        _faobj_order=14/100;

        if upcase(paramcd) in ('MAXTEMP') then
            do;
                _faobj_order=_faobj_order + 1;
                _faobj_label=trim(propcase(FAOBJ));
            end;
        else if upcase(paramcd) in ('MAXSFAT', 'MAXCHIL', 'MAXDEAP', 'MASIRR',
            'MASINSL', ") then
            do;
                _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
                _faobj_order=_faobj_order + 2;
            end;
        else if upcase(paramcd) in ('MAXSHEA') then
            do;
                _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
                _faobj_order=_faobj_order + 2;
            end;
        else if upcase(paramcd) in ('MAXSNV', 'MAXSVOM', 'MAXSNU') then
            do;
                _faobj_order=_faobj_order + 3;
                _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super e}";
            end;
        else if upcase(paramcd) in ('MAXDIAR') then
            do;
                _faobj_order=_faobj_order + 4;

```

```

        _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super f}";
    end;
else if upcase(paramcd) in ('FEVMEDGI', 'PAIMEDGI', 'MEDTFVPN') then
    do;
        _faobj_order=_faobj_order + 99;
        _faobj_label="Use of antipyretic or pain medication"||"(*ESC*){super h}";
    end;
else if upcase(paramcd) in ('OCRASH', 'OCDECAP', 'OCHIVES', 'OCDECSL') then
    do;
        _faobj_order=_faobj_order + 5;
        _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super g}";
    end;
else if upcase(paramcd) in ('MASNMP', 'MAAMP', 'MAXSMP') then
    do;
        _faobj_label=trim("New or worsened muscle pain")||"(*ESC*){super d}";
        _faobj_order=_faobj_order + 6;
    end;
else if upcase(paramcd) in ('MANJP', 'MAXAJP', 'MAXSJP') then
    do;
        _faobj_label=trim("New or worsened joint pain")||"(*ESC*){super d}";
        _faobj_order=_faobj_order + 7;
    end;
else
    do;
        _faobj_order=_faobj_order + 8;
        _faobj_label=trim(propcase(FAOBJ));
    end;
output;
end;

if upcase(paramcd)="MAAMP" then
    do;
        _faobj_order=15/100;

        if upcase(paramcd) in ('MAXTEMP') then
            do;
                _faobj_order=_faobj_order + 1;
                _faobj_label=trim(propcase(FAOBJ));
            end;
        else if upcase(paramcd) in ('MAXSFAT', 'MAXCHIL', 'MAXDEAP', 'MASIRR',
            'MASINSL', ") then
            do;
                _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
                _faobj_order=_faobj_order + 2;
            end;
        else if upcase(paramcd) in ('MAXSHEA') then
            do;
                _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
                _faobj_order=_faobj_order + 2;
            end;
        else if upcase(paramcd) in ('MAXSNV', 'MAXSVOM', 'MAXSNU') then
            do;
                _faobj_order=_faobj_order + 3;
                _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super e}";

```

FDA-CBER-2022-5812-0735682

```

        end;
    else if upcase(paramcd) in ('MAXDIAR') then
        do;
            _faobj_order=_faobj_order + 4;
            _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super f}";
        end;
    else if upcase(paramcd) in ('FEVMEDGI', 'PAIMEDGI', 'MEDTFVPN') then
        do;
            _faobj_order=_faobj_order + 99;
            _faobj_label="Use of antipyretic or pain medication"||"(*ESC*){super h}";
        end;
    else if upcase(paramcd) in ('OCRASH', 'OCDECAP', 'OCHIVES', 'OCDECSL') then
        do;
            _faobj_order=_faobj_order + 5;
            _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super g}";
        end;
    else if upcase(paramcd) in ('MASNMP', 'MAAMP', 'MAXSMP') then
        do;
            _faobj_label=trim("New or worsened muscle pain")||"(*ESC*){super d}";
            _faobj_order=_faobj_order + 6;
        end;
    else if upcase(paramcd) in ('MANJP', 'MAXAJP', 'MAXSJP') then
        do;
            _faobj_label=trim("New or worsened joint pain")||"(*ESC*){super d}";
            _faobj_order=_faobj_order + 7;
        end;
    else
        do;
            _faobj_order=_faobj_order + 8;
            _faobj_label=trim(propcase(FAOBJ));
        end;
    output;
end;

if upcase(paramcd)="MAXSMP" then
    do;
        _faobj_order=16/100;

        if upcase(paramcd) in ('MAXTEMP') then
            do;
                _faobj_order=_faobj_order + 1;
                _faobj_label=trim(propcase(FAOBJ));
            end;
        else if upcase(paramcd) in ('MAXSFAT', 'MAXCHIL', 'MAXDEAP', 'MASIRR',
            'MASINSL', '') then
            do;
                _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
                _faobj_order=_faobj_order + 2;
            end;
        else if upcase(paramcd) in ('MAXSHEA') then
            do;
                _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
                _faobj_order=_faobj_order + 2;
            end;
        end;
    end;
end;

```

```

else if upcase(paramcd) in ('MAXSNV', 'MAXSVOM' 'MAXSNU') then
  do;
    _faobj_order=_faobj_order + 3;
    _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super e}";
  end;
else if upcase(paramcd) in ('MAXDIAR') then
  do;
    _faobj_order=_faobj_order + 4;
    _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super f}";
  end;
else if upcase(paramcd) in ('FEVMEDGI', 'PAIMEDGI', 'MEDTFVPN') then
  do;
    _faobj_order=_faobj_order + 99;
    _faobj_label="Use of antipyretic or pain medication"||"(*ESC*){super h}";
  end;
else if upcase(paramcd) in ('OCRASH', 'OCDECAP', 'OCHIVES', 'OCDECSL') then
  do;
    _faobj_order=_faobj_order + 5;
    _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super g}";
  end;
else if upcase(paramcd) in ('MASNMP', 'MAAMP', 'MAXSMP') then
  do;
    _faobj_label=trim("New or worsened muscle pain")||"(*ESC*){super d}";
    _faobj_order=_faobj_order + 6;
  end;
else if upcase(paramcd) in ('MANJP', 'MAXAJP', 'MAXSJP') then
  do;
    _faobj_label=trim("New or worsened joint pain")||"(*ESC*){super d}";
    _faobj_order=_faobj_order + 7;
  end;
else
  do;
    _faobj_order=_faobj_order + 8;
    _faobj_label=trim(propcase(FAOBJ));
  end;
output;
end;

if upcase(paramcd)="MANJP" then
  do;
    _faobj_order=17/100;

    if upcase(paramcd) in ('MAXTEMP') then
      do;
        _faobj_order=_faobj_order + 1;
        _faobj_label=trim(propcase(FAOBJ));
      end;
    else if upcase(paramcd) in ('MAXSFAT', 'MAXCHIL', 'MAXDEAP', 'MASIRR',
      'MASINSL', ") then
      do;
        _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
        _faobj_order=_faobj_order + 2;
      end;
    else if upcase(paramcd) in ('MAXSHEA') then

```

```

do;
    _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
    _faobj_order=_faobj_order + 2;
end;
else if upcase(paramcd) in ('MAXSNV', 'MAXSVOM', 'MAXSNU') then
do;
    _faobj_order=_faobj_order + 3;
    _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super e}";
end;
else if upcase(paramcd) in ('MAXDIAR') then
do;
    _faobj_order=_faobj_order + 4;
    _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super f}";
end;
else if upcase(paramcd) in ('FEVMEDGI', 'PAIMEDGI', 'MEDTFVPN') then
do;
    _faobj_order=_faobj_order + 99;
    _faobj_label="Use of antipyretic or pain medication"||"(*ESC*){super h}";
end;
else if upcase(paramcd) in ('OCRASH', 'OCDECAP', 'OCHIVES', 'OCDECSL') then
do;
    _faobj_order=_faobj_order + 5;
    _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super g}";
end;
else if upcase(paramcd) in ('MASNMP', 'MAAMP', 'MAXSMP') then
do;
    _faobj_label=trim("New or worsened muscle pain")||"(*ESC*){super d}";
    _faobj_order=_faobj_order + 6;
end;
else if upcase(paramcd) in ('MANJP', 'MAXAJP', 'MAXSJP') then
do;
    _faobj_label=trim("New or worsened joint pain")||"(*ESC*){super d}";
    _faobj_order=_faobj_order + 7;
end;
else
do;
    _faobj_order=_faobj_order + 8;
    _faobj_label=trim(propcase(FAOBJ));
end;
output;
end;

if upcase(paramcd)="MAXAJP" then
do;
    _faobj_order=18/100;

    if upcase(paramcd) in ('MAXTEMP') then
do;
    _faobj_order=_faobj_order + 1;
    _faobj_label=trim(propcase(FAOBJ));
end;
else if upcase(paramcd) in ('MAXSFAT', 'MAXCHIL', 'MAXDEAP', 'MASIRR',
'MASINSL', '') then
do;

```

```

        _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
        _faobj_order=_faobj_order + 2;
    end;
else if upcase(paramcd) in ('MAXSHEA') then
    do;
        _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
        _faobj_order=_faobj_order + 2;
    end;
else if upcase(paramcd) in ('MAXSNV', 'MAXSVOM' 'MAXSNU') then
    do;
        _faobj_order=_faobj_order + 3;
        _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super e}";
    end;
else if upcase(paramcd) in ('MAXDIAR') then
    do;
        _faobj_order=_faobj_order + 4;
        _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super f}";
    end;
else if upcase(paramcd) in ('FEVMEDGI', 'PAIMEDGI', 'MEDTFVPN') then
    do;
        _faobj_order=_faobj_order + 99;
        _faobj_label="Use of antipyretic or pain medication"||"(*ESC*){super h}";
    end;
else if upcase(paramcd) in ('OCRASH', 'OCDECAP', 'OCHIVES', 'OCDECSL') then
    do;
        _faobj_order=_faobj_order + 5;
        _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super g}";
    end;
else if upcase(paramcd) in ('MASNMP', 'MAAMP', 'MAXSMP') then
    do;
        _faobj_label=trim("New or worsened muscle pain")||"(*ESC*){super d}";
        _faobj_order=_faobj_order + 6;
    end;
else if upcase(paramcd) in ('MANJP', 'MAXAJP', 'MAXSJP') then
    do;
        _faobj_label=trim("New or worsened joint pain")||"(*ESC*){super d}";
        _faobj_order=_faobj_order + 7;
    end;
else
    do;
        _faobj_order=_faobj_order + 8;
        _faobj_label=trim(propcase(FAOBJ));
    end;
output;
end;

if upcase(paramcd)="OCDECAP" then
    do;
        _faobj_order=19/100;

        if upcase(paramcd) in ('MAXTEMP') then
            do;
                _faobj_order=_faobj_order + 1;
                _faobj_label=trim(propcase(FAOBJ));
            end;
        end;
    end;

```

```

        end;
    else if upcase(paramcd) in ('MAXSFAT', 'MAXCHIL', 'MAXDEAP', 'MASIRR',
        'MASINSL', '') then
        do;
            _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
            _faobj_order=_faobj_order + 2;
        end;
    else if upcase(paramcd) in ('MAXSHEA') then
        do;
            _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
            _faobj_order=_faobj_order + 2;
        end;
    else if upcase(paramcd) in ('MAXSNV', 'MAXSVOM', 'MAXSNU') then
        do;
            _faobj_order=_faobj_order + 3;
            _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super e}";
        end;
    else if upcase(paramcd) in ('MAXDIAR') then
        do;
            _faobj_order=_faobj_order + 4;
            _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super f}";
        end;
    else if upcase(paramcd) in ('FEVMEDGI', 'PAIMEDGI', 'MEDTFVPN') then
        do;
            _faobj_order=_faobj_order + 99;
            _faobj_label="Use of antipyretic or pain medication"||"(*ESC*){super h}";
        end;
    else if upcase(paramcd) in ('OCRASH', 'OCDECAP', 'OCHIVES', 'OCDECSL') then
        do;
            _faobj_order=_faobj_order + 5;
            _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super g}";
        end;
    else if upcase(paramcd) in ('MASNMP', 'MAAMP', 'MAXSMP') then
        do;
            _faobj_label=trim("New or worsened muscle pain")||"(*ESC*){super d}";
            _faobj_order=_faobj_order + 6;
        end;
    else if upcase(paramcd) in ('MANJP', 'MAXAJP', 'MAXSJP') then
        do;
            _faobj_label=trim("New or worsened joint pain")||"(*ESC*){super d}";
            _faobj_order=_faobj_order + 7;
        end;
    else
        do;
            _faobj_order=_faobj_order + 8;
            _faobj_label=trim(propcase(FAOBJ));
        end;
    output;
end;

if upcase(paramcd)="MAXSJP" then
    do;
        _faobj_order=20/100;
    end;

```



```

if upcase(paramcd) in ('MAXTEMP') then
  do;
    _faobj_order=_faobj_order + 1;
    _faobj_label=trim(propcase(FAOBJ));
  end;
else if upcase(paramcd) in ('MAXSFAT', 'MAXCHIL', 'MAXDEAP', 'MASIRR',
'MASINSL', '') then
  do;
    _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
    _faobj_order=_faobj_order + 2;
  end;
else if upcase(paramcd) in ('MAXSHEA') then
  do;
    _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
    _faobj_order=_faobj_order + 2;
  end;
else if upcase(paramcd) in ('MAXSNV', 'MAXSVOM', 'MAXSNU') then
  do;
    _faobj_order=_faobj_order + 3;
    _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super e}";
  end;
else if upcase(paramcd) in ('MAXDIAR') then
  do;
    _faobj_order=_faobj_order + 4;
    _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super f}";
  end;
else if upcase(paramcd) in ('FEVMEDGI', 'PAIMEDGI', 'MEDTFVPN') then
  do;
    _faobj_order=_faobj_order + 99;
    _faobj_label="Use of antipyretic or pain medication"||"(*ESC*){super h}";
  end;
else if upcase(paramcd) in ('OCRASH', 'OCDECAP', 'OCHIVES', 'OCDECSL') then
  do;
    _faobj_order=_faobj_order + 5;
    _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super g}";
  end;
else if upcase(paramcd) in ('MASNMP', 'MAAMP', 'MAXSMP') then
  do;
    _faobj_label=trim("New or worsened muscle pain")||"(*ESC*){super d}";
    _faobj_order=_faobj_order + 6;
  end;
else if upcase(paramcd) in ('MANJP', 'MAXAJP', 'MAXSJP') then
  do;
    _faobj_label=trim("New or worsened joint pain")||"(*ESC*){super d}";
    _faobj_order=_faobj_order + 7;
  end;
else
  do;
    _faobj_order=_faobj_order + 8;
    _faobj_label=trim(propcase(FAOBJ));
  end;
output;
end;

```

```

if upcase(paramcd)="MASIRR" then
  do;
    _faobj_order=21/100;

    if upcase(paramcd) in ('MAXTEMP') then
      do;
        _faobj_order=_faobj_order + 1;
        _faobj_label=trim(propcase(FAOBJ));
      end;
    else if upcase(paramcd) in ('MAXSFAT', 'MAXCHIL', 'MAXDEAP', 'MASIRR',
      'MASINSL', '') then
      do;
        _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
        _faobj_order=_faobj_order + 2;
      end;
    else if upcase(paramcd) in ('MAXSHEA') then
      do;
        _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
        _faobj_order=_faobj_order + 2;
      end;
    else if upcase(paramcd) in ('MAXSNV', 'MAXSVOM', 'MAXSNU') then
      do;
        _faobj_order=_faobj_order + 3;
        _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super e}";
      end;
    else if upcase(paramcd) in ('MAXDIAR') then
      do;
        _faobj_order=_faobj_order + 4;
        _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super f}";
      end;
    else if upcase(paramcd) in ('FEVMEDGI', 'PAIMEDGI', 'MEDTFVPN') then
      do;
        _faobj_order=_faobj_order + 99;
        _faobj_label="Use of antipyretic or pain medication"||"(*ESC*){super h}";
      end;
    else if upcase(paramcd) in ('OCRASH', 'OCDECAP', 'OCHIVES', 'OCDECSL') then
      do;
        _faobj_order=_faobj_order + 5;
        _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super g}";
      end;
    else if upcase(paramcd) in ('MASNMP', 'MAAMP', 'MAXSMP') then
      do;
        _faobj_label=trim("New or worsened muscle pain")||"(*ESC*){super d}";
        _faobj_order=_faobj_order + 6;
      end;
    else if upcase(paramcd) in ('MANJP', 'MAXAJP', 'MAXSJP') then
      do;
        _faobj_label=trim("New or worsened joint pain")||"(*ESC*){super d}";
        _faobj_order=_faobj_order + 7;
      end;
    else
      do;
        _faobj_order=_faobj_order + 8;
        _faobj_label=trim(propcase(FAOBJ));
      end;

```

```

        end;
    output;
end;

if upcase(paramcd)="MASINSL" then
do;
    _faobj_order=22/100;

    if upcase(paramcd) in ('MAXTEMP') then
        do;
            _faobj_order=_faobj_order + 1;
            _faobj_label=trim(propcase(FAOBJ));
        end;
    else if upcase(paramcd) in ('MAXSFAT', 'MAXCHIL', 'MAXDEAP', 'MASIRR',
        'MASINSL', '') then
        do;
            _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
            _faobj_order=_faobj_order + 2;
        end;
    else if upcase(paramcd) in ('MAXSHEA') then
        do;
            _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
            _faobj_order=_faobj_order + 2;
        end;
    else if upcase(paramcd) in ('MAXSNV', 'MAXSVOM', 'MAXSNU') then
        do;
            _faobj_order=_faobj_order + 3;
            _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super e}";
        end;
    else if upcase(paramcd) in ('MAXDIAR') then
        do;
            _faobj_order=_faobj_order + 4;
            _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super f}";
        end;
    else if upcase(paramcd) in ('FEVMEDGI', 'PAIMEDGI', 'MEDTFVPN') then
        do;
            _faobj_order=_faobj_order + 99;
            _faobj_label="Use of antipyretic or pain medication"||"(*ESC*){super h}";
        end;
    else if upcase(paramcd) in ('OCRASH', 'OCDECAP', 'OCHIVES', 'OCDECSL') then
        do;
            _faobj_order=_faobj_order + 5;
            _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super g}";
        end;
    else if upcase(paramcd) in ('MASNMP', 'MAAMP', 'MAXSMP') then
        do;
            _faobj_label=trim("New or worsened muscle pain")||"(*ESC*){super d}";
            _faobj_order=_faobj_order + 6;
        end;
    else if upcase(paramcd) in ('MANJP', 'MAXAJP', 'MAXSJP') then
        do;
            _faobj_label=trim("New or worsened joint pain")||"(*ESC*){super d}";
            _faobj_order=_faobj_order + 7;
        end;
end;

```

```

else
  do;
    _faobj_order=_faobj_order + 8;
    _faobj_label=trim(propcase(FAOBJ));
  end;
output;
end;

if upcase(paramcd)="MAXSNV" then
  do;
    _faobj_order=23/100;

    if upcase(paramcd) in ('MAXTEMP') then
      do;
        _faobj_order=_faobj_order + 1;
        _faobj_label=trim(propcase(FAOBJ));
      end;
    else if upcase(paramcd) in ('MAXSFAT', 'MAXCHIL', 'MAXDEAP', 'MASIRR',
      'MASINSL', '') then
      do;
        _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
        _faobj_order=_faobj_order + 2;
      end;
    else if upcase(paramcd) in ('MAXSHEA') then
      do;
        _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
        _faobj_order=_faobj_order + 2;
      end;
    else if upcase(paramcd) in ('MAXSNV', 'MAXSVOM', 'MAXSNU') then
      do;
        _faobj_order=_faobj_order + 3;
        _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super e}";
      end;
    else if upcase(paramcd) in ('MAXDIAR') then
      do;
        _faobj_order=_faobj_order + 4;
        _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super f}";
      end;
    else if upcase(paramcd) in ('FEVMEDGI', 'PAIMEDGI', 'MEDTFVPN') then
      do;
        _faobj_order=_faobj_order + 99;
        _faobj_label="Use of antipyretic or pain medication"||"(*ESC*){super h}";
      end;
    else if upcase(paramcd) in ('OCRASH', 'OCDECAP', 'OCHIVES', 'OCDECSL') then
      do;
        _faobj_order=_faobj_order + 5;
        _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super g}";
      end;
    else if upcase(paramcd) in ('MASNMP', 'MAAMP', 'MAXSMP') then
      do;
        _faobj_label=trim("New or worsened muscle pain")||"(*ESC*){super d}";
        _faobj_order=_faobj_order + 6;
      end;
    else if upcase(paramcd) in ('MANJP', 'MAXAJP', 'MAXSJP') then

```

FDA-CBER-2022-5812-0735691

```

do;
    _faobj_label=trim("New or worsened joint pain")||"(*ESC*){super d}";
    _faobj_order=_faobj_order + 7;
end;
else
do;
    _faobj_order=_faobj_order + 8;
    _faobj_label=trim(propcase(FAOBJ));
end;
output;
end;

if upcase(paramcd)="ANY" then
do;
    _faobj_order=24/100;

    if upcase(paramcd) in ('MAXTEMP') then
do;
    _faobj_order=_faobj_order + 1;
    _faobj_label=trim(propcase(FAOBJ));
end;
else if upcase(paramcd) in ('MAXSFAT', 'MAXCHIL', 'MAXDEAP', 'MASIRR',
'MASINSL', '') then
do;
    _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
    _faobj_order=_faobj_order + 2;
end;
else if upcase(paramcd) in ('MAXSHEA') then
do;
    _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super d}";
    _faobj_order=_faobj_order + 2;
end;
else if upcase(paramcd) in ('MAXSNV', 'MAXSVOM', 'MAXSNU') then
do;
    _faobj_order=_faobj_order + 3;
    _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super e}";
end;
else if upcase(paramcd) in ('MAXDIAR') then
do;
    _faobj_order=_faobj_order + 4;
    _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super f}";
end;
else if upcase(paramcd) in ('FEVMEDGI', 'PAIMEDGI', 'MEDTFVPN') then
do;
    _faobj_order=_faobj_order + 99;
    _faobj_label="Use of antipyretic or pain medication"||"(*ESC*){super h}";
end;
else if upcase(paramcd) in ('OCRASH', 'OCDECAP', 'OCHIVES', 'OCDECSL') then
do;
    _faobj_order=_faobj_order + 5;
    _faobj_label=trim(propcase(FAOBJ))||"(*ESC*){super g}";
end;
else if upcase(paramcd) in ('MASNMP', 'MAAMP', 'MAXSMP') then
do;

```

```

        _faobj_label=trim("New or worsened muscle pain")||"(*ESC*){super d}";
        _faobj_order=_faobj_order + 6;
    end;
else if upcase(paramcd) in ('MANJP', 'MAXAJP', 'MAXSJP') then
    do;
        _faobj_label=trim("New or worsened joint pain")||"(*ESC*){super d}";
        _faobj_order=_faobj_order + 7;
    end;
else
    do;
        _faobj_order=_faobj_order + 8;
        _faobj_label=trim(propcase(FAOBJ));
    end;
output;
end;
run;

```

```

data _rash _hives _decsleep _medic _a_dsin _app;
set _a_dsin;

```

```

if paramcd='OCRASH' then
    do;

        if AVALC='Y' then
            AVAL=1;

        if AVALC='N' then
            AVAL=0;
        output _rash;
    end;

```

```

if paramcd='OCHIVES' then
    do;

        if AVALC='Y' then
            AVAL=1;

        if AVALC='N' then
            AVAL=0;
        output _hives;
    end;

```

```

if paramcd='OCDECSL' then
    do;

        if AVALC='Y' then
            AVAL=1;

        if AVALC='N' then
            AVAL=0;
        output _decsleep;
    end;
else if paramcd in ('FEVMEDGI', 'PAIMEDGI', 'MEDTFVPN') then
    do;

```

```

        if AVALC='Y' then
            AVAL=1;

        if AVALC='N' then
            AVAL=0;

        if knowvfl='Y' then
            _knowvfl=1;

        if knowvfl='N' then
            _knowvfl=0;

        if missing(aval) then
            aval=0;
        PARAMCD='PAIMEDGI';
        output _medic;
    end;
else if paramcd in ('OCDECAP') then
    do;

        if AVALC='Y' then
            AVAL=1;

        if AVALC='N' then
            AVAL=0;
        output _app;
    end;
else
    do;
        output _a_dsin;
    end;
run;

proc sort data=_rash;
    by atptref newtrt usubjid aval;
run;

data _rash;
    set _rash;
    by atptref newtrt usubjid aval;

    if last.usubjid;
run;

proc sort data=_hives;
    by atptref newtrt usubjid aval;
run;

data _hives;
    set _hives;
    by atptref newtrt usubjid aval;

    if last.usubjid;

```

```

run;

proc sort data=_decsleep;
    by atptref newtrt usubjid aval;
run;

data _decsleep;
    set _decsleep;
    by atptref newtrt usubjid aval;

    if last.usubjid;
run;

proc sort data=_app;
    by atptref newtrt usubjid aval;
run;

data _app;
    set _app;
    by atptref newtrt usubjid aval;

    if last.usubjid;
run;

proc sort data=_medic;
    by atptref newtrt usubjid aval _knowvfl;
run;

data _medic;
    set _medic;
    by atptref newtrt usubjid aval _knowvfl;

    if last.usubjid;
run;

data _a_dsin;
    set _a_dsin _rash _medic _app _hives _decsleep;

    if upcase(paramcd) in ('MAXTEMP') then
        do;

            if ftemcatn > 0 then
                do;
                    AVAL=FTEMCATN;
                    AVALC=FTEMCAT;
                end;
            else if ftemcatn=0 then
                do;
                    AVALC='NONE';
                    AVAL=0;
                end;
            end;
        end;
run;

```



```

proc sort data=_a_dsin;
    by newtrt faobj usubjid atptref descending eventfl descending aval;
run;

data _a_dsin;
    set _a_dsin;
    by newtrt faobj usubjid atptref descending eventfl descending aval;

    if first.atptref;
run;

data _a_any;
    set _a_dsin;

    if PARAMCD not in ('PAIMEDGI');
    FAOBJ='Any systemic event';
    PARAMCD='ANY';
    _faobj_order=90;
    _faobj_label=trim(FAOBJ)||"(*ESC*){super g}";
run;

proc sort data=_a_any;
    by newtrt usubjid atptref descending aval;
run;

data _a_any;
    set _a_any;
    by newtrt usubjid atptref descending aval;

    if first.atptref then
        _any_flg=1;
    else
        _any_flg=0;
run;

data _a_dsin;
    set _a_dsin _a_any;
run;

data anysev;
    set _a_dsin;

    if ^missing(aval) and FATESTCD in ('MAXSEV', 'MAXTEMP') then
        do;

            if aval=0 then
                do;
                    ex_none_flg=1;
                end;
            else
                do;
                    ex_none_flg=0;
                end;
            AVALC='ANY';

```

```

        AVAL=0;
        output;
    end;
run;

proc sql;
    create table fev_bign as select distinct newtrt, usubjid, paramcd, knowvfl ,
        atptref from _a_dsin where upcase(paramcd) in ('MAXTEMP');
quit;

data fev_bign;
    set fev_bign;
    DENOMFL=0;
    output;
    DENOMFL=1;
    output;
    DENOMFL=2;
    output;
    DENOMFL=3;
    output;
    DENOMFL=4;
    output;
run;

proc sort data=fev_bign;
    by newtrt usubjid paramcd knowvfl atptref;
quit;

proc sort data=_a_dsin;
    by newtrt usubjid paramcd knowvfl atptref;
quit;

data _a_dsin;
    merge _a_dsin fev_bign;
    by newtrt usubjid paramcd knowvfl atptref;
run;

data _a_dsin;
    set _a_dsin anysev;

    if upcase(AVALC) in ('N', 'NONE') then
        aval=999;
run;

proc sql;
    create table _bigN as select distinct newtrt, usubjid, paramcd, knowvfl ,
        atptref from _a_dsin where fatestcd='MAXSEV' and paramcd ne "ANY";
    create table _bigN_any as select distinct newtrt, usubjid, paramcd, knowvfl ,
        atptref from _a_dsin where paramcd="ANY";
quit;

data _bigN;
    set _bigN _bigN_any;
    DENOMFL=0;

```

```

output;
DENOMFL=1;
output;
DENOMFL=2;
output;
DENOMFL=3;
output;
DENOMFL=4;
output;
run;

proc sort data=_a_dsin;
  by newtrt usubjid paramcd knowvfl atptref;
quit;

proc sort data=_bigN;
  by newtrt usubjid paramcd knowvfl atptref;
quit;

data _a_dsin;
  merge _a_dsin _bigN;
  by newtrt usubjid paramcd knowvfl atptref;

  if paramcd in ('OCRASH', 'PAIMEDGI', 'OCDECAP', 'OCHIVES', 'OCDECSL')
    and ^missing(aval) then
    do;
      denomfl=1;
    end;
run;

data _dsin_terms(keep=paramcd _faobj_order _faobj_label);
  set _a_dsin;
run;

proc sort data=_dsin_terms nodupkey;
  by paramcd _faobj_label;
run;

proc sort data=_dsin_terms out=grp(keep=paramcd _faobj_label) nodupkey;
  by _faobj_order;
run;

proc sort data=g_adsl_dsin out=_ds1;
  by usubjid;
run;

proc sort data=_a_dsin out=_ds2;
  by usubjid;
run;

data final;
  merge _ds1(in=d1) _ds2(in=d2);
  by usubjid;

```

```

    if d2;
run;
proc sort data=final;
    by newtrt usubjid;
run;

data final;
    set final;
    length denomflc $150.;

    if knowvfl='Y' then
        _knowvfl=1;

    if avalc not in ('ANY', 'NONE') then
        ex_none_flg=0;

    if paramcd='MAXTEMP' then
        do;

            if aval=0 then
                do;
                    avalc=(*ESC*){unicode 2265}38.0||"(*ESC*){Unicode 2103}";
                end;

            if aval=1 then
                do;
                    avalc=(*ESC*){unicode 2265}38.0||"(*ESC*){Unicode 00B0}C" || '||'to||' || '38.4||"(*ESC*)
{Unicode 00B0}C";
                end;

            if aval=2 then
                do;
                    avalc='>38.4||"(*ESC*){Unicode 00B0}C" || '||'to||' || '38.9||"(*ESC*){Unicode 00B0}C";
                end;

            if aval=3 then
                do;
                    avalc='>38.9||"(*ESC*){Unicode 00B0}C" || '||'to||' || '40.0||"(*ESC*){Unicode 00B0}C";
                end;

            if aval=4 then
                do;
                    avalc='>40.0||"(*ESC*){Unicode 00B0}C";
                end;

            if denomfl=0 then
                do;
                    denomflc=(*ESC*){unicode 2265}38.0||"(*ESC*){Unicode 00B0}C";
                end;

            if denomfl=1 then
                do;
                    denomflc=(*ESC*){unicode 2265}38.0||"(*ESC*){Unicode 00B0}C" || '||'to||' || '38.4||"
(*ESC*){Unicode 00B0}C";

```

```

        end;

    if denomfl=2 then
        do;
            denomflc='>38.4' || "(*ESC*){Unicode 00B0}C" || ' ' || 'to' || ' ' || '38.9' || "(*ESC*){Unicode 00B0}C";
        end;

    if denomfl=3 then
        do;
            denomflc='>38.9' || "(*ESC*){Unicode 00B0}C" || ' ' || 'to' || ' ' || '40.0' || "(*ESC*){Unicode 00B0}C";
        end;

    if denomfl=4 then
        do;
            denomflc='>40.0' || "(*ESC*){Unicode 00B0}C";
        end;

    end;

run;

data final;
    set final;

    if paramcd in ('ANY') and ^missing(aval) then
        do;
            denomfl=1;

            if aval > 0 then
                aval=1;
        end;

run;

```

```

*****
* Specification 4                                     *;
* Create a template dataset                           *;
*****

```

```

*-----;
* Initialize structure for _BASETEMPLATE dataset. ;
*-----;

```

```

data _basetemplate(compress=no);
    length _varname $8 _cvalue $30 _direct $20 _vrlabel $200 _rwlabel
        _colabel $800 _datatyp $5 _module $8 _pr_lbl $ 200;
    array _c _character_;
    delete;

run;

```

```

data _data1;
    set final;
    where (NEWTRTN is not missing);

```

```

run;

proc sort data=_data1;
    by NEWTRTN USUBJID;
run;

data _data1;
    retain _trt 0;
    length _str $200;
    _datasrt=1;
    set _data1 end=eof;
    by NEWTRTN USUBJID;
    drop _str;
    _str='';
    _lastby=1;
    _dummyby=0;

    if first.NEWTRTN then
        do;

            if not missing(NEWTRTN) then
                do;
                    _trt=_trt + 1;
                end;

            *-----;
            * Generate _STR as the treatment label ;
            *-----;
            _str=NEWTRT;
            *-----;
            * Update _TRTLB&n with generated treatment label ;
            *-----;

            if _trt > 0 then
                call symput('_trtlb'||compress(put(_trt, 4.)), trim(left(_str)));
        end;
run;

*-----;
* Handle sub-group N=xxx/sub-group analysis request ;
*-----;

proc sql noprint;
    select count(unique AGEGR4) into :_subGrpN from _data1 where AGEGR4 is not
        missing;
    create table _subGrpData as select distinct _trt, AGEGR4, count(distinct
        USUBJID) as _subGrpCnt from _data1 where AGEGR4 is not missing group by _trt,
        AGEGR4;
quit;

proc sql noprint;
    create table _subGrpDataVH as select distinct 9999 as _trt, AGEGR4,
        count(distinct USUBJID) as _subGrpCnt from _data1 where AGEGR4 is not missing
        group by AGEGR4;
quit;

```

```

data _subGrpData;
  length _cat $100;
  set _subGrpData;
  by _trt;
  _cat=AGEGR4;
run;

data _trtframe;
  _trt=ifN(1 eq 3, 9999, 1);
  output;
  _trt=ifN(2 eq 3, 9999, 2);
  output;
run;

proc sql noprint;
  create table _fullSubGrp as select * from (select distinct _trt from
    _trtframe), (select distinct _cat from _subGrpData) order by _trt, _cat;
quit;

data _fullSubGrp;
  set _fullSubGrp;
  by _trt _cat;

  if first._trt then
    _subcat=0;
    _subcat + 1;

  if _subcat=2 + 1 then
    _subcat=9999;
run;

data _subGrpData _tmpsubGrpData;
  merge _fullSubGrp _subGrpData;
  by _trt _cat;
  length _colabel $200;
  _colabel=_cat;

  if _subGrpCnt=. then
    _subGrpCnt=0;
run;

/* data _subGrpDataVH; */
/*   length _cat $100; */
/*   set _subGrpDataVH; */
/*   by _trt; */
/*   _cat=AGEGR4; */
/* run; */
/* */
/* proc sql noprint; */
/*   create table _fullSubGrpVH as select * from (select distinct _trt from */
/*     _subGrpDataVH), (select distinct _cat from _subgrpdatavh) order by _trt, _cat; */
/* */
/* data _fullSubGrpVH; */

```

```

/* set _fullSubGrpVH; */
/* by _trt _cat; */
/* */
/* if first._trt then */
/*     _subcat=0; */
/*     _subcat + 1; */
/* */
/* if _subcat=2 + 1 then */
/*     _subcat=9999; */
/* run; */
/* */
/* data _subGrpDataVH _tmpsubGrpDataVH; */
/* merge _subGrpDataVH _fullSubGrpVH; */
/* by _trt _cat; */
/* length _colabel $200; */
/* _colabel=_cat; */
/* */
/* if _subGrpCnt=. then */
/*     _subGrpCnt=0; */
/* run; */

*-----;
* Generate a dataset containing all by-variables ;
*-----;
proc sort data=_data1 out=_bydat1(keep=_datasrt ATPREFN ATPREF
    _dummyby) nodupkey;
    by _datasrt ATPREFN;
run;

data _bydat1;
    set _bydat1 end=eof;
    by _datasrt ATPREFN;
    retain _preby 0;
    drop _preby ATPREF;
    length _bylab1- _bylab1 $100;
    retain _byvar1- _byvar1 0 _bylen1- _bylen1 0 _bylab1- _bylab1;

    if first.ATPREFN then
        do;
            _byvar1 + 1;
            _bylab1=ATPREF;
            _bylen1=length(_bylab1);
        end;
    output;

    if last.ATPREFN then
        do;
            if _byvar1 > _preby then
                _preby=_byvar1;
            call symput("_preby1", compress(put(_preby, 4.)));
        end;

```



```

        if eof then
            do;
                call symput("_preby1", compress(put(_byvar1, 4.)));

                if 1=0 then
                    output;
            end;
run;

data _bydat1;
    set _bydat1;
    by _datasrt;
    length _bycol _byindnt $50 _bylast $10;
    _bycol="1";
    _byindnt="0";
    _bylast=" ";
run;

proc sort data=_bydat1;
    by _datasrt ATPTREFN;
run;

proc sort data=_data1 out=_data1;
    by _datasrt ATPTREFN;
run;

*-----;
* Merge calculated by variables back into _DATAn dataset. ;
*-----;

data _data1;
    merge _bydat1(keep=_datasrt _byvar1 ATPTREFN) _data1(in=_b);
    by _datasrt ATPTREFN;

    if _b;
run;

proc sort data=_data1;
    by _datasrt _byvar1;
run;

*****
***;
* Specification 5                                     *;
* 1) Count N: number of subjects with any e-diary data reported after Vaccination 1      *;
* 2) Count n and %: number of subjects with the specified characteristic and proportion    *;
* 3) Calculate 95% CI for %: exact 2-sided CI based on the Clopper and Pearson method      *;
*****
***;

*****;
* Specification 5.1: Statistics for Fever category      *;
*****;

```

```
* Specification 5.1.1: Count denominator (N) *;  
*****;
```

```
data _anal1;  
  length DENOMFL 8;  
  length _cat $100;  
  set _data1;  
  where AGEGR4 is not missing;  
  where same and DENOMFL is not missing;  
  _blksrt=1;  
  _cnt=1;  
  _cat=AGEGR4;  
  
  if _trt <=0 then  
    delete;  
  output;  
run;  
  
proc sort data=_anal1;  
  by _datasrt _byvar1 _blksrt DENOMFL _trt _cat;  
run;  
  
proc sort data=_anal1 out=_catby1(keep=_byvar1) nodupkey;  
  by _byvar1;  
  where paramcd eq upcase("Maxtemp");  
run;  
  
data _temp1;  
  set _anal1;  
  output;  
run;  
  
proc sort data=_temp1 out=_temp91 nodupkey;  
  by _datasrt _byvar1 _blksrt _cat DENOMFL _trt usubjid;  
  where paramcd eq upcase("Maxtemp");  
run;  
  
proc freq data=_temp91 noprint;  
  format DENOMFL;  
  tables _datasrt*_byvar1*_blksrt*_cat * DENOMFL * _trt / sparse norow nocol  
    nopercnt out=_pct1(drop=percent);  
run;  
  
proc freq data=_pct1 noprint;  
  where DENOMFL ne 9999;  
  weight count;  
  tables _datasrt*_byvar1*_cat * _trt / sparse noprint out=_denom1(drop=percent);  
run;  
  
data _denomf1;  
  length _cat $100;  
  _datasrt=1;  
  set _catby1(keep=_byvar1);
```

```

* All treatment groups ;
_trt1=0;
_trt2=0;
* _CAT is the subgroup variable ;
_cat="12-15 Years ";
output;
_cat="16-25 Years ";
output;
run;

proc transpose data=_denom1 out=_denomin1(drop=_name__label_) prefix=_trt;
  by _datasrt _byvar1 _cat;
  var count;
  id _trt;
run;

proc sort data=_anal1 out=_catlab1(keep=DENOMFL denomflc) nodupkey;
  by DENOMFL denomflc;
  where paramcd eq upcase("Maxtemp");
run;

data _frame1;
  _datasrt=1;
  set _catby1(keep=_byvar1);
  _blcksrt=1;
  length DENOMFL 8;
  length _cat $100;
  length _catLabl $150;
  _catLabl=' ';
  _catLabl="(*ESC*){unicode 2265}38.0(*ESC*){Unicode 00B0}C";
  _trt=1;
  DENOMFL=0;
  _catord=1;
  _subcat=1;
  _cat="12-15 Years ";
  output;
  _subcat=2;
  _cat="16-25 Years ";
  output;
  _trt=2;
  DENOMFL=0;
  _catord=1;
  _subcat=1;
  _cat="12-15 Years ";
  output;
  _subcat=2;
  _cat="16-25 Years ";
  output;
  _catLabl="(*ESC*){unicode 2265}38.0(*ESC*){Unicode 00B0}C to 38.4(*ESC*){Unicode 00B0}C";
  _trt=1;
  DENOMFL=1;
  _catord=2;
  _subcat=1;

```

```

_cat="12-15 Years ";
output;
_subcat=2;
_cat="16-25 Years ";
output;
_trt=2;
DENOMFL=1;
_catord=2;
_subcat=1;
_cat="12-15 Years ";
output;
_subcat=2;
_cat="16-25 Years ";
output;
_catLabl=">38.4(*ESC*){Unicode 00B0}C to 38.9(*ESC*){Unicode 00B0}C";
_trt=1;
DENOMFL=2;
_catord=3;
_subcat=1;
_cat="12-15 Years ";
output;
_subcat=2;
_cat="16-25 Years ";
output;
_trt=2;
DENOMFL=2;
_catord=3;
_subcat=1;
_cat="12-15 Years ";
output;
_subcat=2;
_cat="16-25 Years ";
output;
_catLabl=">38.9(*ESC*){Unicode 00B0}C to 40.0(*ESC*){Unicode 00B0}C";
_trt=1;
DENOMFL=3;
_catord=4;
_subcat=1;
_cat="12-15 Years ";
output;
_subcat=2;
_cat="16-25 Years ";
output;
_trt=2;
DENOMFL=3;
_catord=4;
_subcat=1;
_cat="12-15 Years ";
output;
_subcat=2;
_cat="16-25 Years ";
output;
_catLabl=">40.0(*ESC*){Unicode 00B0}C";
_trt=1;

```

```

DENOMFL=4;
_catord=5;
_subcat=1;
_cat="12-15 Years ";
output;
_subcat=2;
_cat="16-25 Years ";
output;
_trt=2;
DENOMFL=4;
_catord=5;
_subcat=1;
_cat="12-15 Years ";
output;
_subcat=2;
_cat="16-25 Years ";
output;
run;

proc sort data=_frame1;
  by _datasrt _byvar1 _blcksrt _cat DENOMFL _trt;
run;

proc sort data=_pct1;
  by _datasrt _byvar1 _blcksrt _cat DENOMFL _trt;
run;

data _pct1;
  merge _frame1(in=_inframe) _pct1;
  by _datasrt _byvar1 _blcksrt _cat DENOMFL _trt;

  if _inframe;

  if count=. then
    count=0;
run;

proc sort data=_pct1;
  by _datasrt _byvar1 _blcksrt DENOMFL;
run;

data _miss1(keep=_datasrt _byvar1 _blcksrt DENOMFL totcount);
  set _pct1;
  where DENOMFL=9998;
  retain totcount;
  by _datasrt _byvar1 _blcksrt DENOMFL;

  if first.DENOMFL then
    totcount=0;
  totcount=totcount+count;

  if last.DENOMFL;
run;

```

```

data _pct1(drop=totcount);
  merge _pct1 _miss1;
  by _datasrt _byvar1 _blcksrt DENOMFL;

  if totcount=0 then
    delete;
run;

proc sort data=_denomf1;
  by _datasrt _byvar1 _cat;
run;

proc sort data=_denomin1;
  by _datasrt _byvar1 _cat;
run;

data _denomin1;
  merge _denomf1(in=_inframe) _denomin1;
  by _datasrt _byvar1 _cat;

  if _inframe;
  _blcksrt=1;
run;

proc sort data=_pct1;
  by _datasrt _byvar1 _cat;
run;

data _pct1;
  if 0 then
    set _basetemplate;
  merge _denomin1(in=_a) _pct1;
  by _datasrt _byvar1 _cat;

  if _a;
  _varname="DENOMFL ";
  _vrlabel="Fever ";
  _rwlabel=_catLbl;

  if DENOMFL=9998 then
    do;
      _rwlabel="Missing ";
      _catord=9998;
    end;
  else if DENOMFL=9999 then
    do;
      _rwlabel="Total ";
      _catord=9999;
    end;

  if _catord=. then
    _catord=9997;
run;

```

```

proc sort data=_pct1;
  by _datasrt _byvar1 _blcksrt _catord DENOMFL _trt _cat;
run;

data _base1;
  length _catlabl $200;
  set _pct1 end=eof;
  by _datasrt _byvar1 _blcksrt _catord DENOMFL _trt _cat;
  retain _rowsrt 0 _rowmax 0;
  array _trcnt(*) _trt1-_trt3;
  drop _rowmax _cpct;
  length _cpct $100;
  _cpct=' ';
  _module='mcatstat';

  if count > . then
    _cvalue=put(count, 5.);
  else
    _cvalue=put(0, 5.);

  if length(_cvalue) < 5 then
    do;
      *-----;
      * Put character A0x at right most character to pad text;
      *-----;
      substr(_cvalue, 5, 1)='A0'x;
    end;

  if first._byvar1 then
    _rowsrt=0;

  if first.DENOMFL then
    do;
      _rowsrt=_rowsrt + 1;
      _rowmax=max(_rowsrt, _rowmax);
    end;
  _datatype='data';
  _indent=0;
  _dptindt=0;
  _vorder=1;
  _rowjump=1;

  if upcase(_rwlabel)='_NONE_' then
    _rwlabel=' ';
  _indent=8;
  _dptindt=0;

  if _trt=2 +1 then
    _trt=9999;

  if eof then
    call symput('_rowsrt', compress(put(_rowmax, 4.)));
  _direct="TOP ";
  _p=0;

```

```

run;

*****
* Specification 5.1.2: Count n and percentage (%) for individual severity    *;
*****

data _anal2;
  length AVAL 8;
  length _cat $100;
  set _data1;
  where AGEGR4 is not missing;
  where same and AVAL is not missing;
  _blcksrt=1;
  _cnt=1;
  _cat=AGEGR4;

  if _trt <=0 then
    delete;
  output;
run;

proc sort data=_anal2;
  by _datasrt _byvar1 _blcksrt AVAL _trt _cat;
run;

proc sort data=_anal2 out=_catby2(keep=_byvar1) nodupkey;
  by _byvar1;
  where paramcd eq upcase("Maxtemp") and ex_none_flg=0 and eventfl="Y";
run;

data _temp2;
  set _anal2;
  output;
run;

proc sort data=_temp2 out=_temp92 nodupkey;
  by _datasrt _byvar1 _blcksrt _cat AVAL _trt usubjid;
  where paramcd eq upcase("Maxtemp") and ex_none_flg=0 and eventfl="Y";
  ;
run;

proc freq data=_temp92 noprint;
  format AVAL;
  tables _datasrt*_byvar1*_blcksrt*_cat * AVAL * _trt / sparse norow nocol
    nopercnt out=_pct2(drop=percent);
run;

proc sort data=_temp2 out=_analcnt2 nodupkey;
  by _datasrt _byvar1 _cat _trt USUBJID;
  where paramcd eq upcase("Maxtemp") and knowvfl eq 'Y';
run;

```



```

proc freq data=_analcnt2 noprint;
  tables _datasrt*_byvar1*_cat * _trt / sparse noprint out=_denom2(drop=percent);
run;

data _denomf2;
  length _cat $100;
  _datasrt=1;
  set _catby2(keep=_byvar1);
  * All treatment groups ;
  _trt1=0;
  _trt2=0;
  * _CAT is the subgroup variable ;
  _cat="12-15 Years ";
  output;
  _cat="16-25 Years ";
  output;
run;

proc transpose data=_denom2 out=_denomin2(drop=_name__label_) prefix=_trt;
  by _datasrt _byvar1 _cat;
  var count;
  id _trt;
run;

proc sort data=_anal2 out=_catlab2(keep=AVAL denomflc) nodupkey;
  by AVAL denomflc;
  where paramcd eq upcase("Maxtemp") and ex_none_flg=0 and eventfl="Y";
run;

data _frame2;
  _datasrt=1;
  set _catby2(keep=_byvar1);
  _blcksrt=1;
  length AVAL 8;
  length _cat $100;
  length _catLabl $150;
  _catLabl=' ';
  _catLabl="";
  _trt=1;
  AVAL=0;
  _catord=1;
  _subcat=1;
  _cat="12-15 Years ";
  output;
  _subcat=2;
  _cat="16-25 Years ";
  output;
  _trt=2;
  AVAL=0;
  _catord=1;
  _subcat=1;
  _cat="12-15 Years ";
  output;
  _subcat=2;

```

```

    _cat="16-25 Years ";
output;
    _catLabl=">40.0(*ESC*){Unicode 00B0}C
";
    _trt=1;
AVAL=1;
    _catord=2;
    _subcat=1;
    _cat="12-15 Years ";
output;
    _subcat=2;
    _cat="16-25 Years ";
output;
    _trt=2;
AVAL=1;
    _catord=2;
    _subcat=1;
    _cat="12-15 Years ";
output;
    _subcat=2;
    _cat="16-25 Years ";
output;
    _catLabl=">40.0(*ESC*){Unicode 00B0}C
";
    _trt=1;
AVAL=2;
    _catord=3;
    _subcat=1;
    _cat="12-15 Years ";
output;
    _subcat=2;
    _cat="16-25 Years ";
output;
    _trt=2;
AVAL=2;
    _catord=3;
    _subcat=1;
    _cat="12-15 Years ";
output;
    _subcat=2;
    _cat="16-25 Years ";
output;
    _catLabl=">40.0(*ESC*){Unicode 00B0}C
";
    _trt=1;
AVAL=3;
    _catord=4;
    _subcat=1;
    _cat="12-15 Years ";
output;
    _subcat=2;
    _cat="16-25 Years ";
output;
    _trt=2;

```

```

    AVAL=3;
    _catord=4;
    _subcat=1;
    _cat="12-15 Years ";
    output;
    _subcat=2;
    _cat="16-25 Years ";
    output;
    _catLbl=">40.0(*ESC*){Unicode 00B0}C
";
    _trt=1;
    AVAL=4;
    _catord=5;
    _subcat=1;
    _cat="12-15 Years ";
    output;
    _subcat=2;
    _cat="16-25 Years ";
    output;
    _trt=2;
    AVAL=4;
    _catord=5;
    _subcat=1;
    _cat="12-15 Years ";
    output;
    _subcat=2;
    _cat="16-25 Years ";
    output;
run;

proc sort data=_frame2;
    by _datasrt _byvar1 _blcksrt _cat AVAL _trt;
run;

proc sort data=_pct2;
    by _datasrt _byvar1 _blcksrt _cat AVAL _trt;
run;

data _pct2;
    merge _frame2(in=_inframe) _pct2;
    by _datasrt _byvar1 _blcksrt _cat AVAL _trt;

    if _inframe;

    if count=. then
        count=0;
run;

proc sort data=_pct2;
    by _datasrt _byvar1 _blcksrt AVAL;
run;

data _miss2(keep=_datasrt _byvar1 _blcksrt AVAL totcount);
    set _pct2;

```

```

where AVAL=9998;
retain totcount;
by _datasrt _byvar1 _blcksrt AVAL;

if first.AVAL then
    totcount=0;
totcount=totcount+count;

if last.AVAL;
run;

data _pct2(drop=totcount);
merge _pct2 _miss2;
by _datasrt _byvar1 _blcksrt AVAL;

if totcount=0 then
    delete;
run;

proc sort data=_denomf2;
by _datasrt _byvar1 _cat;
run;

proc sort data=_denomin2;
by _datasrt _byvar1 _cat;
run;

data _denomin2;
merge _denomf2(in=_inframe) _denomin2;
by _datasrt _byvar1 _cat;

if _inframe;
    _blcksrt=1;
run;

proc sort data=_pct2;
by _datasrt _byvar1 _cat;
run;

data _pct2;
if 0 then
    set _basemplate;
merge _denomin2(in=_a) _pct2;
by _datasrt _byvar1 _cat;

if _a;
    _varname="AVAL ";
    _vrlabel=" ";
    _rwlabel=_catLbl;

if AVAL=9998 then
    do;
        _rwlabel="Missing ";
        _catord=9998;

```

```

        end;
    else if AVAL=9999 then
        do;
            _rwlabel="Total ";
            _catord=9999;
        end;

    if _catord=. then
        _catord=9997;

run;

proc sort data=_pct2;
    by _datasrt _byvar1 _blcksrt _catord AVAL _trt _cat;
run;

data _base2;
    length _catlabl $200;
    set _pct2 end=eof;
    by _datasrt _byvar1 _blcksrt _catord AVAL _trt _cat;
    retain _rowsrt 5 _rowmax 0;
    array _trtcnt(*) _trt1-_trt3;
    drop _rowmax _cpct;
    length _cpct $100;
    _cpct=' ';
    _module='mcatstat';

    if count > . then
        _cvalue=put(count, 5.);
    else
        _cvalue=put(0, 5.);
    *-----;
    * Format percent to append to display value in _CVALUE ;
    *-----;

    if _trt ne . then
        do;

            if _trtcnt(_trt) > 0 then
                do;
                    percent=count / _trtcnt(_trt) * 100;

                    if percent > 0 then
                        do;

                            if round(percent, 0.1) GE 0.1 then
                                _cpct="(*ESC*){nbspspace 1}("||strip(put(percent, 5.1))||")";
                            else
                                _cpct="(*ESC*){nbspspace 1}(0.0)";
                            _cvalue=trim(_cvalue)||_cpct;
                        end;
                    end;

                end;

            end;

        end;

    if length(_cvalue) < 13 then

```

```

do;
    *-----;
    * Put character A0x at right most character to pad text;
    *-----;
    substr(_cvalue, 13, 1)='A0'x;
end;

if first._byvar1 then
    _rowsrt=5;

if first.AVAL then
do;
    _rowsrt=_rowsrt + 1;
    _rowmax=max(_rowsrt, _rowmax);
end;
_datatyp='data';
_indent=0;
_dptindt=0;
_vorder=1;
_rowjump=1;

if upcase(_rwlabel)='_NONE_' then
    _rwlabel=' ';
_indent=8;
_dptindt=0;

if _trt=2 +1 then
    _trt=9999;

if eof then
    call symput('_rowsrt', compress(put(_rowmax, 4.)));
_direct="TOP ";
_p=2;
run;

data _base2;
set _base2;
length _cvalue2 $30.;
_cvalue2=strip(tranwrd(_cvalue, 'A0'x, ""));
_cvalue21=strip(scan(_cvalue, 1, '('));
_cvalue22=compress(scan(_cvalue, 2, '(', ')'));
run;

data _base1;
set _base1;
drop _trt1 _trt2 count;
run;

proc sort data=_base1;
by _datasrt _byvar1 _cat _trt;
run;

data _base1;
set _base1;

```

```

if _trt=1 then
  do;
    _trt1=input(_cvalue, ??best.);
  end;

if _trt=2 then
  do;
    _trt2=input(_cvalue, ??best.);
  end;

run;

```

```

proc sort data=_base2(keep=_datasrt _trt _cvalue2 _cvalue21 _cvalue22 _cat
  _byvar1 count);
  by _datasrt _byvar1 _cat _trt;
quit;

```

```

data _base2;
  merge _base1(in=a) _base2(in=b);
  by _datasrt _byvar1 _cat _trt;

```

```

if a;

if a and not b then
  do;
    _cvalue2="0";
    _cvalue21="0";
  end;

```

```

if compress(_cvalue2)="0" then
  _cvalue22=put(0, 5.1);

```

```

if compress(_cvalue)="0" then
  do;
    _cvalue2="NA";
    _cvalue21="NA";
    _cvalue22="NA";
  end;

```

```
run;
```

```

data _base1;
  set _base1;
  delete;

```

```
run;
```

```

*****
* Specification 5.1.3: Calculate 95% CI for observed proportion      *;
*****

```

```

data _cnp _tmp_cnp;
  set _base2;

```

```

if count=. then
  count=0;

```

```

indc=1;
output _cnp;
indc=2;

if _trt=1 then
  do;
    count=_trt1 - count;
  end;

if _trt=2 then
  do;
    count=_trt2 - count;
  end;
output _cnp;

if indc=2 and count=0 then
  output _tmp_cnp;
run;

proc sort data=_cnp;
  by _byvar1 _cat _rowsrt _rwlabel _trt;
run;

proc sort nodupkey data=_tmp_cnp(keep=_byvar1 _cat _rowsrt _rwlabel _trt);
  by _byvar1 _cat _rowsrt _rwlabel _trt;
run;

*****
* Call proc freq procedure to calculate CI for observed proportion          *;
*****

proc freq data=_cnp noprint;
  by _byvar1 _cat _rowsrt _rwlabel _trt;
  table indc/binomial alpha=0.05;
  output out=obsprop binomial;
  weight count;
run;

data obsprop;
  merge obsprop _tmp_cnp(in=a);
  by _byvar1 _cat _rowsrt _rwlabel _trt;

  if _bin_=1 and not a then
    do;
      xl_bin_=1 - xu_bin;
      xu_bin_=1 - xl_bin;
    end;
  else
    do;
      xl_bin_=xl_bin;
      xu_bin_=xu_bin;
    end;
run;

```



```

data cnpobsprop1(keep=_byvar1 _cat _rowsrt _rwlabel _trt cnp_ci);
  set obsprop;
  by _byvar1 _cat _rowsrt _rwlabel _trt;
  cnp_ci=(' || compress(put(xl_bin_ * 100, 5.1))
    || '(*ESC*){nbspace 1}' || compress(put(xu_bin_ * 100, 5.1)) || ');
  label cnp_ci='95% CI';
run;

proc datasets lib=work nolist gennum=all;
  delete _cnp obsprop;
run;

proc sort data=_base2;
  by _byvar1 _cat _rowsrt _rwlabel _trt;
run;

proc sort data=cnpobsprop1;
  by _byvar1 _cat _rowsrt _rwlabel _trt;
run;

data _base2;
  merge _base2(in=a) cnpobsprop1;
  by _byvar1 _cat _rowsrt _rwlabel _trt;

  if a;

  if compress(_cvalue)="0" then
    do;
      cnp_ci="NE";
    end;
run;

*****
* Specification 5.2: Statistics for Fatigue category *;
*****
* Specification 5.2.1: Count denominator (N) *;
*****

data _anal3;
  set _anal1;
  _blcksrt=2;
run;

proc sort data=_anal3;
  by _datasrt _byvar1 _blcksrt DENOMFL _trt _cat;
run;

proc sort data=_anal3 out=_catby3(keep=_byvar1) nodupkey;
  by _byvar1;
  where paramcd eq upcase("Maxsfat");
run;

data _temp3;
  set _anal3;

```

```

output;
run;

proc sort data=_temp3 out=_temp93 nodupkey;
  by _datasrt _byvar1 _blcksrt _cat DENOMFL _trt usubjid;
  where paramcd eq upcase("Maxsfat");
run;

proc freq data=_temp93 noprint;
  format DENOMFL;
  tables _datasrt*_byvar1*_blcksrt*_cat * DENOMFL * _trt / sparse norow nocol
    nopercnt out=_pct3(drop=percent);
run;

proc freq data=_pct3 noprint;
  where DENOMFL ne 9999;
  weight count;
  tables _datasrt*_byvar1*_cat * _trt / sparse noprint out=_denom3(drop=percent);
run;

data _denomf3;
  length _cat $100;
  _datasrt=1;
  set _catby3(keep=_byvar1);
  * All treatment groups ;
  _trt1=0;
  _trt2=0;
  * _CAT is the subgroup variable ;
  _cat="12-15 Years ";
  output;
  _cat="16-25 Years ";
  output;
run;

proc transpose data=_denom3 out=_denomin3(drop=_name __label_) prefix=_trt;
  by _datasrt _byvar1 _cat;
  var count;
  id _trt;
run;

data _frame3;
  set _frame1;
  _blcksrt=2;
run;

proc sort data=_frame3;
  by _datasrt _byvar1 _blcksrt _cat DENOMFL _trt;
run;

proc sort data=_pct3;
  by _datasrt _byvar1 _blcksrt _cat DENOMFL _trt;
run;

```

```

data _pct3;
  merge _frame3(in=_inframe) _pct3;
  by _datasrt _byvar1 _blcksrt _cat DENOMFL _trt;

  if _inframe;

  if count=. then
    count=0;
run;

proc sort data=_pct3;
  by _datasrt _byvar1 _blcksrt DENOMFL;
run;

data _miss3(keep=_datasrt _byvar1 _blcksrt DENOMFL totcount);
  set _pct3;
  where DENOMFL=9998;
  retain totcount;
  by _datasrt _byvar1 _blcksrt DENOMFL;

  if first.DENOMFL then
    totcount=0;
  totcount=totcount+count;

  if last.DENOMFL;
run;

data _pct3(drop=totcount);
  merge _pct3 _miss3;
  by _datasrt _byvar1 _blcksrt DENOMFL;

  if totcount=0 then
    delete;
run;

proc sort data=_denomf3;
  by _datasrt _byvar1 _cat;
run;

proc sort data=_denomin3;
  by _datasrt _byvar1 _cat;
run;

data _denomin3;
  merge _denomf3(in=_inframe) _denomin3;
  by _datasrt _byvar1 _cat;

  if _inframe;
  _blcksrt=2;
run;

proc sort data=_pct3;
  by _datasrt _byvar1 _cat;

```

```

run;

data _pct3;
  if 0 then
    set _basetemplate;
  merge _denomin3(in=_a) _pct3;
  by _datasrt _byvar1 _cat;

  if _a;
  _varname="DENOMFL ";
  _vrlabel="Fatigue(*ESC*){super d} ";
  _rwlabel=put(DENOMFL, sev.);

  if DENOMFL=9998 then
    do;
      _rwlabel="Missing ";
      _catord=9998;
    end;
  else if DENOMFL=9999 then
    do;
      _rwlabel="Total ";
      _catord=9999;
    end;

  if _catord=. then
    _catord=9997;

run;

proc sort data=_pct3;
  by _datasrt _byvar1 _blcksrt _catord DENOMFL _trt _cat;
run;

data _base3;
  length _catlabl $200;
  set _pct3 end=eof;
  by _datasrt _byvar1 _blcksrt _catord DENOMFL _trt _cat;
  retain _rowsrt 0 _rowmax 0;
  array _trtcnt(*) _trt1- _trt3;
  drop _rowmax _cpct;
  length _cpct $100;
  _cpct='';
  _module='mcatstat';

  if count > . then
    _cvalue=put(count, 5.);
  else
    _cvalue=put(0, 5.);

  if length(_cvalue) < 5 then
    do;
      *-----;
      * Put character A0x at right most character to pad text;
      *-----;
      substr(_cvalue, 5, 1)='A0'x;
    end;

```

```

        end;

    if first._byvar1 then
        _rowsrt=0;

    if first.DENOMFL then
        do;
            _rowsrt=_rowsrt + 1;
            _rowmax=max(_rowsrt, _rowmax);
        end;
    _datatyp='data';
    _indent=0;
    _dptindt=0;
    _vorder=1;
    _rowjump=1;

    if upcase(_rwlabel)='_NONE_' then
        _rwlabel=' ';
    _indent=8;
    _dptindt=0;

    if _trt=2 +1 then
        _trt=9999;

    if eof then
        call symput('_rowsrt', compress(put(_rowmax, 4.)));
    _direct="TOP ";
    _p=0;
run;

*****
* Specification 5.2.2: Count n and percentage (%) for individual severity    *;
*****

data _anal4;
    set _anal2;
    _blcksrt=2;
run;

proc sort data=_anal4;
    by _datasrt _byvar1 _blcksrt AVAL _trt _cat;
run;

proc sort data=_anal4 out=_catby4(keep=_byvar1) nodupkey;
    by _byvar1;
    where paramcd eq upcase("Maxsfat") and ex_none_flg=0 and eventfl="Y";
;
run;

data _temp4;
    set _anal4;
    output;
run;

```

```

proc sort data=_temp4 out=_temp94 nodupkey;
  by _datasrt _byvar1 _blcksrt _cat AVAL _trt usubjid;
  where paramcd eq upcase("Maxsfat") and ex_none_flg=0 and eventfl="Y";

proc freq data=_temp94 noprint;
  format AVAL;
  tables _datasrt*_byvar1*_blcksrt*_cat * AVAL * _trt / sparse norow nocol
    nopercnt out=_pct4(drop=percent);
run;

proc sort data=_temp4 out=_analcnt4 nodupkey;
  by _datasrt _byvar1 _cat _trt USUBJID;
  where paramcd eq upcase("Maxsfat") and knowvfl eq 'Y';
run;

proc freq data=_analcnt4 noprint;
  tables _datasrt*_byvar1*_cat * _trt / sparse noprint out=_denom4(drop=percent);
run;

data _denomf4;
  length _cat $100;
  _datasrt=1;
  set _catby4(keep=_byvar1);
  * All treatment groups ;
  _trt1=0;
  _trt2=0;
  * _CAT is the subgroup variable ;
  _cat="12-15 Years ";
  output;
  _cat="16-25 Years ";
  output;
run;

proc transpose data=_denom4 out=_denomin4(drop=_name _label_) prefix=_trt;
  by _datasrt _byvar1 _cat;
  var count;
  id _trt;
run;

data _frame4;
  set _frame2;
  _blcksrt=2;
run;

proc sort data=_frame4;
  by _datasrt _byvar1 _blcksrt _cat AVAL _trt;
run;

proc sort data=_pct4;
  by _datasrt _byvar1 _blcksrt _cat AVAL _trt;
run;

data _pct4;
  merge _frame4(in=_inframe) _pct4;

```

```

    by _datasrt _byvar1 _blcksrt _cat AVAL _trt;

    if _inframe;

    if count=. then
        count=0;
run;

proc sort data=_pct4;
    by _datasrt _byvar1 _blcksrt AVAL;
run;

data _miss4(keep=_datasrt _byvar1 _blcksrt AVAL totcount);
    set _pct4;
    where AVAL=9998;
    retain totcount;
    by _datasrt _byvar1 _blcksrt AVAL;

    if first.AVAL then
        totcount=0;
    totcount=totcount+count;

    if last.AVAL;
run;

data _pct4(drop=totcount);
    merge _pct4 _miss4;
    by _datasrt _byvar1 _blcksrt AVAL;

    if totcount=0 then
        delete;
run;

proc sort data=_denomf4;
    by _datasrt _byvar1 _cat;
run;

proc sort data=_denomin4;
    by _datasrt _byvar1 _cat;
run;

data _denomin4;
    merge _denomf4(in=_inframe) _denomin4;
    by _datasrt _byvar1 _cat;

    if _inframe;
    _blcksrt=2;
run;

proc sort data=_pct4;
    by _datasrt _byvar1 _cat;
run;

data _pct4;

```

```

if 0 then
    set _basetemplate;
merge _denomin4(in=_a) _pct4;
by _datasrt _byvar1 _cat;

if _a;
    _varname="AVAL ";
    _vrlabel=" ";
    _rwlabel=put(AVAL, sev.);

if AVAL=9998 then
    do;
        _rwlabel="Missing ";
        _catord=9998;
    end;
else if AVAL=9999 then
    do;
        _rwlabel="Total ";
        _catord=9999;
    end;

if _catord=. then
    _catord=9997;

run;

proc sort data=_pct4;
    by _datasrt _byvar1 _blcksrt _catord AVAL _trt _cat;
run;

data _base4;
    length _catlabl $200;
    set _pct4 end=eof;
    by _datasrt _byvar1 _blcksrt _catord AVAL _trt _cat;
    retain _rowsrt 5 _rowmax 0;
    array _trtcnt(*) _trt1-_trt3;
    drop _rowmax _cpct;
    length _cpct $100;
    _cpct='';
    _module='mcatstat';

if count > . then
    _cvalue=put(count, 5.);
else
    _cvalue=put(0, 5.);
*-----;
* Format percent to append to display value in _CVALUE ;
*-----;

if _trt ne . then
    do;

        if _trtcnt(_trt) > 0 then
            do;
                percent=count / _trtcnt(_trt) * 100;

```



```

        if percent > 0 then
            do;

                if round(percent, 0.1) GE 0.1 then
                    _cpct="(*ESC*){nbspspace 1}("||strip(put(percent, 5.1))||")";
                else
                    _cpct="(*ESC*){nbspspace 1}(0.0)";
                _cvalue=trim(_cvalue)||_cpct;
            end;
        end;

    end;

if length(_cvalue) < 13 then
    do;
        *-----;
        * Put character A0x at right most character to pad text;
        *-----;
        substr(_cvalue, 13, 1)='A0'x;
    end;

if first._byvar1 then
    _rowsrt=6;

if first.AVAL then
    do;
        _rowsrt=_rowsrt + 1;
        _rowmax=max(_rowsrt, _rowmax);
    end;
_datatyp='data';
_indent=0;
_dptindt=0;
_vorder=1;
_rowjump=1;

if upcase(_rwlabel)='_NONE_' then
    _rwlabel=' ';
_indent=8;
_dptindt=0;

if _trt=2 +1 then
    _trt=9999;

if eof then
    call symput('_rowsrt', compress(put(_rowmax, 4.)));
_direct="TOP ";
_p=2;

run;

data _base4;
set _base4;
length _cvalue2 $30.;
_cvalue2=strip(tranwrd(_cvalue, 'A0'x, ""));
_cvalue21=strip(scan(_cvalue, 1, '('));

```

```

        _cvalue22=compress(scan(_cvalue, 2, '(', ')'));
run;

data _base3;
    set _base3;
    drop _trt1 _trt2 count;
run;

proc sort data=_base3;
    by _datasrt _byvar1 _cat _trt;
run;

data _base3;
    set _base3;

    if _trt=1 then
        do;
            _trt1=input(_cvalue, ??best.);
        end;

    if _trt=2 then
        do;
            _trt2=input(_cvalue, ??best.);
        end;
run;

proc sort data=_base4(keep=_datasrt _trt _cvalue2 _cvalue21 _cvalue22 _cat
    _byvar1 count);
    by _datasrt _byvar1 _cat _trt;
quit;

data _base4;
    merge _base3(in=a) _base4(in=b);
    by _datasrt _byvar1 _cat _trt;

    if a;

    if a and not b then
        do;
            _cvalue2="0";
            _cvalue21="0";
        end;

    if compress(_cvalue2)="0" then
        _cvalue22=put(0, 5.1);

    if compress(_cvalue)="0" then
        do;
            _cvalue2="NA";
            _cvalue21="NA";
            _cvalue22="NA";
        end;
run;

```

```

data _base3;
    set _base3;
    delete;
run;

*****
* Specification 5.2.3: Calculate 95% CI for observed proportion          *;
*****

data _cnp_tmp_cnp;
    set _base4;

    if count=. then
        count=0;
    indc=1;
    output _cnp;
    indc=2;

    if _trt=1 then
        do;
            count=_trt1 - count;
        end;

    if _trt=2 then
        do;
            count=_trt2 - count;
        end;
    output _cnp;

    if indc=2 and count=0 then
        output _tmp_cnp;
run;

proc sort data=_cnp;
    by _byvar1 _cat _rowsrt _rwlabel _trt;
run;

proc sort nodupkey data=_tmp_cnp(keep=_byvar1 _cat _rowsrt _rwlabel _trt);
    by _byvar1 _cat _rowsrt _rwlabel _trt;
run;

*****
* Call proc freq procedure to calculate CI for observed proportion      *;
*****

proc freq data=_cnp noprint;
    by _byvar1 _cat _rowsrt _rwlabel _trt;
    table indc/binomial alpha=0.05;
    output out=obsprop binomial;
    weight count;
run;

data obsprop;
    merge obsprop _tmp_cnp(in=a);
    by _byvar1 _cat _rowsrt _rwlabel _trt;

```

```

if _bin_=1 and not a then
  do;
    xl_bin_=1 - xu_bin;
    xu_bin_=1 - xl_bin;
  end;
else
  do;
    xl_bin_=xl_bin;
    xu_bin_=xu_bin;
  end;
run;

data cnpobsprop1(keep=_byvar1 _cat _rowsrt _rwlabel _trt cnp_ci);
  set obsprop;
  by _byvar1 _cat _rowsrt _rwlabel _trt;
  cnp_ci=(' ' || compress(put(xl_bin_ * 100, 5.1))
    || '(*ESC*){nbspspace 1}' || compress(put(xu_bin_ * 100, 5.1)) || ');
  label cnp_ci='95% CI';
run;

proc datasets lib=work nolist gennum=all;
  delete _cnp obsprop;
run;

proc sort data=_base4;
  by _byvar1 _cat _rowsrt _rwlabel _trt;
run;

proc sort data=cnpobsprop1;
  by _byvar1 _cat _rowsrt _rwlabel _trt;
run;

data _base4;
  merge _base4(in=a) cnpobsprop1;
  by _byvar1 _cat _rowsrt _rwlabel _trt;

  if a;

  if compress(_cvalue)="0" then
    do;
      cnp_ci="NE";
    end;
run;

*****
* Specification 5.3: Statistics for Headache category          *;
*****
* Specification 5.3.1: Count denominator (N)                  *;
*****

data _anal5;
  set _anal1;
  _blcksrt=3;

```

```

run;

proc sort data=_anal5;
  by _datasrt _byvar1 _blcksrt DENOMFL _trt _cat;
run;

proc sort data=_anal5 out=_catby5(keep=_byvar1) nodupkey;
  by _byvar1;
  where paramcd eq upcase("Maxshea");
run;

data _temp5;
  set _anal5;
  output;
run;

proc sort data=_temp5 out=_temp95 nodupkey;
  by _datasrt _byvar1 _blcksrt _cat DENOMFL _trt usubjid;
  where paramcd eq upcase("Maxshea");
run;

proc freq data=_temp95 noprint;
  format DENOMFL;
  tables _datasrt*_byvar1*_blcksrt*_cat * DENOMFL * _trt / sparse norow nocol
    nopercnt out=_pct5(drop=percent);
run;

proc freq data=_pct5 noprint;
  where DENOMFL ne 9999;
  weight count;
  tables _datasrt*_byvar1*_cat * _trt / sparse noprint out=_denom5(drop=percent);
run;

data _denomf5;
  length _cat $100;
  _datasrt=1;
  set _catby5(keep=_byvar1);
  * All treatment groups ;
  _trt1=0;
  _trt2=0;
  * _CAT is the subgroup variable ;
  _cat="12-15 Years ";
  output;
  _cat="16-25 Years ";
  output;
run;

proc transpose data=_denom5 out=_denomin5(drop=_name __label_) prefix=_trt;
  by _datasrt _byvar1 _cat;
  var count;
  id _trt;
run;

data _frame5;

```

```

    set _frame3;
    _blcksrt=3;
run;

proc sort data=_frame5;
    by _datasrt _byvar1 _blcksrt _cat DENOMFL _trt;
run;

proc sort data=_pct5;
    by _datasrt _byvar1 _blcksrt _cat DENOMFL _trt;
run;

data _pct5;
    merge _frame5(in=_inframe) _pct5;
    by _datasrt _byvar1 _blcksrt _cat DENOMFL _trt;

    if _inframe;

    if count=. then
        count=0;
run;

proc sort data=_pct5;
    by _datasrt _byvar1 _blcksrt DENOMFL;
run;

data _miss5(keep=_datasrt _byvar1 _blcksrt DENOMFL totcount);
    set _pct5;
    where DENOMFL=9998;
    retain totcount;
    by _datasrt _byvar1 _blcksrt DENOMFL;

    if first.DENOMFL then
        totcount=0;
    totcount=totcount+count;

    if last.DENOMFL;
run;

data _pct5(drop=totcount);
    merge _pct5 _miss5;
    by _datasrt _byvar1 _blcksrt DENOMFL;

    if totcount=0 then
        delete;
run;

proc sort data=_denomf5;
    by _datasrt _byvar1 _cat;
run;

proc sort data=_denomin5;
    by _datasrt _byvar1 _cat;
run;

```

```

data _denomin5;
  merge _denomf5(in=_inframe) _denomin5;
  by _datasrt _byvar1 _cat;

  if _inframe;
  _blcksrt=3;
run;

proc sort data=_pct5;
  by _datasrt _byvar1 _cat;
run;

data _pct5;
  if 0 then
    set _basetemplate;
  merge _denomin5(in=_a) _pct5;
  by _datasrt _byvar1 _cat;

  if _a;
  _varname="DENOMFL ";
  _vrlabel="Headache(*ESC*){super d} ";
  _rwlabel=put(DENOMFL, sev.);

  if DENOMFL=9998 then
    do;
      _rwlabel="Missing ";
      _catord=9998;
    end;
  else if DENOMFL=9999 then
    do;
      _rwlabel="Total ";
      _catord=9999;
    end;

  if _catord=. then
    _catord=9997;
run;

proc sort data=_pct5;
  by _datasrt _byvar1 _blcksrt _catord DENOMFL _trt _cat;
run;

data _base5;
  length _catlabl $200;
  set _pct5 end=eof;
  by _datasrt _byvar1 _blcksrt _catord DENOMFL _trt _cat;
  retain _rowsrt 0 _rowmax 0;
  array _trtcnt(*) _trt1-_trt3;
  drop _rowmax _cpct;
  length _cpct $100;
  _cpct='';
  _module='mcatstat';

```

```

if count > . then
  _cvalue=put(count, 5.);
else
  _cvalue=put(0, 5.);

if length(_cvalue) < 5 then
  do;
    *-----;
    * Put character A0x at right most character to pad text;
    *-----;
    substr(_cvalue, 5, 1)='A0'x;
  end;

if first._byvar1 then
  _rowsrt=0;

if first.DENOMFL then
  do;
    _rowsrt=_rowsrt + 1;
    _rowmax=max(_rowsrt, _rowmax);
  end;
_datatyp='data';
_indent=0;
_dptindt=0;
_vorder=1;
_rowjump=1;

if upcase(_rlabel)='_NONE_' then
  _rlabel=' ';
_indent=8;
_dptindt=0;

if _trt=2 +1 then
  _trt=9999;

if eof then
  call symput('_rowsrt', compress(put(_rowmax, 4.)));
_direct="TOP ";
_p=0;

run;

*****
* Specification 5.3.2: Count n and percentage (%) for individual severity   *;
*****;

data _anal6;
  set _anal2;
  _blksrt=3;

run;

proc sort data=_anal6;
  by _datasrt _byvar1 _blksrt AVAL _trt _cat;

run;

```



```

proc sort data=_anal6 out=_catby6(keep=_byvar1) nodupkey;
  by _byvar1;
  where paramcd eq upcase("Maxshea") and ex_none_flg=0 and eventfl="Y";
run;

data _temp6;
  set _anal6;
  output;
run;

proc sort data=_temp6 out=_temp96 nodupkey;
  by _datasrt _byvar1 _blcksrt _cat AVAL _trt usubjid;
  where paramcd eq upcase("Maxshea") and ex_none_flg=0 and eventfl="Y";
run;

proc freq data=_temp96 noprint;
  format AVAL;
  tables _datasrt*_byvar1*_blcksrt*_cat * AVAL * _trt / sparse norow nocol
    nopercnt out=_pct6(drop=percent);
run;

proc sort data=_temp6 out=_analcnt6 nodupkey;
  by _datasrt _byvar1 _cat _trt USUBJID;
  where paramcd eq upcase("Maxshea") and knowvfl eq 'Y';
run;

proc freq data=_analcnt6 noprint;
  tables _datasrt*_byvar1*_cat * _trt / sparse noprint out=_denom6(drop=percent);
run;

data _denomf6;
  length _cat $100;
  _datasrt=1;
  set _catby6(keep=_byvar1);
  * All treatment groups ;
  _trt1=0;
  _trt2=0;
  * _CAT is the subgroup variable ;
  _cat="12-15 Years ";
  output;
  _cat="16-25 Years ";
  output;
run;

proc transpose data=_denom6 out=_denomin6(drop=_name__label_) prefix=_trt;
  by _datasrt _byvar1 _cat;
  var count;
  id _trt;
run;

data _frame6;
  set _frame4;
  _blcksrt=3;
run;

```

```

proc sort data=_frame6;
    by _datasrt _byvar1 _blcksrt _cat AVAL _trt;
run;

proc sort data=_pct6;
    by _datasrt _byvar1 _blcksrt _cat AVAL _trt;
run;

data _pct6;
    merge _frame6(in=_inframe) _pct6;
    by _datasrt _byvar1 _blcksrt _cat AVAL _trt;

    if _inframe;

    if count=. then
        count=0;
run;

proc sort data=_pct6;
    by _datasrt _byvar1 _blcksrt AVAL;
run;

data _miss6(keep=_datasrt _byvar1 _blcksrt AVAL totcount);
    set _pct6;
    where AVAL=9998;
    retain totcount;
    by _datasrt _byvar1 _blcksrt AVAL;

    if first.AVAL then
        totcount=0;
    totcount=totcount+count;

    if last.AVAL;
run;

data _pct6(drop=totcount);
    merge _pct6 _miss6;
    by _datasrt _byvar1 _blcksrt AVAL;

    if totcount=0 then
        delete;
run;

proc sort data=_denomf6;
    by _datasrt _byvar1 _cat;
run;

proc sort data=_denomin6;
    by _datasrt _byvar1 _cat;
run;

data _denomin6;
    merge _denomf6(in=_inframe) _denomin6;

```

```

    by _datasrt _byvar1 _cat;

    if _inframe;
    _blcksrt=3;
run;

proc sort data=_pct6;
    by _datasrt _byvar1 _cat;
run;

data _pct6;
    if 0 then
        set _basetemplate;
    merge _denomin6(in=_a) _pct6;
    by _datasrt _byvar1 _cat;

    if _a;
    _varname="AVAL ";
    _vrlabel=" ";
    _rwlabel=put(AVAL, sev.);

    if AVAL=9998 then
        do;
            _rwlabel="Missing ";
            _catord=9998;
        end;
    else if AVAL=9999 then
        do;
            _rwlabel="Total ";
            _catord=9999;
        end;

    if _catord=. then
        _catord=9997;
run;

proc sort data=_pct6;
    by _datasrt _byvar1 _blcksrt _catord AVAL _trt _cat;
run;

data _base6;
    length _catlabl $200;
    set _pct6 end=eof;
    by _datasrt _byvar1 _blcksrt _catord AVAL _trt _cat;
    retain _rowsrt 5 _rowmax 0;
    array _trcnt(*) _trt1-_trt3;
    drop _rowmax _cpct;
    length _cpct $100;
    _cpct='';
    _module='mcatstat';

    if count > . then
        _cvalue=put(count, 5.);
    else

```

```

    _cvalue=put(0, 5.);
*-----;
* Format percent to append to display value in _CVALUE ;
*-----;

if _trt ne . then
    do;

        if _trtcnt(_trt) > 0 then
            do;
                percent=count / _trtcnt(_trt) * 100;

                if percent > 0 then
                    do;

                        if round(percent, 0.1) GE 0.1 then
                            _cpct="(*ESC*){nbspspace 1}("||strip(put(percent, 5.1))||")";
                        else
                            _cpct="(*ESC*){nbspspace 1}(0.0)";
                        _cvalue=trim(_cvalue)||_cpct;
                    end;
                end;
            end;
        end;

    if length(_cvalue) < 13 then
        do;
            *-----;
            * Put character A0x at right most character to pad text;
            *-----;
            substr(_cvalue, 13, 1)='A0'x;
        end;

    if first._byvar1 then
        _rowsrt=6;

    if first.AVAL then
        do;
            _rowsrt=_rowsrt + 1;
            _rowmax=max(_rowsrt, _rowmax);
        end;
    _datatyp='data';
    _indent=0;
    _dptindt=0;
    _vorder=1;
    _rowjump=1;

    if upcase(_rwlabel)='_NONE_' then
        _rwlabel=' ';
    _indent=8;
    _dptindt=0;

    if _trt=2+1 then
        _trt=9999;

```

```

    if eof then
        call symput('_rowsrt', compress(put(_rowmax, 4.)));
    _direct="TOP ";
    _p=2;
run;

data _base6;
    set _base6;
    length _cvalue2 $30.;
    _cvalue2=strip(tranwrd(_cvalue, 'A0'x, ""));
    _cvalue21=strip(scan(_cvalue, 1, '('));
    _cvalue22=compress(scan(_cvalue, 2, '('), ''));
run;

data _base5;
    set _base5;
    drop _trt1 _trt2 count;
run;

proc sort data=_base5;
    by _datasrt _byvar1 _cat _trt;
run;

data _base5;
    set _base5;

    if _trt=1 then
        do;
            _trt1=input(_cvalue, ??best.);
        end;

    if _trt=2 then
        do;
            _trt2=input(_cvalue, ??best.);
        end;
run;

proc sort data=_base6(keep=_datasrt _trt _cvalue2 _cvalue21 _cvalue22 _cat
    _byvar1 count);
    by _datasrt _byvar1 _cat _trt;
quit;

data _base6;
    merge _base5(in=a) _base6(in=b);
    by _datasrt _byvar1 _cat _trt;

    if a;

    if a and not b then
        do;
            _cvalue2="0";
            _cvalue21="0";
        end;

```

```

if compress(_cvalue2)="0" then
  _cvalue22=put(0, 5.1);

if compress(_cvalue)="0" then
  do;
    _cvalue2="NA";
    _cvalue21="NA";
    _cvalue22="NA";
  end;
run;

data _base5;
  set _base5;
  delete;
run;

*****
* Specification 5.3.3: Calculate 95% CI for observed proportion      *;
*****

data _cnp _tmp_cnp;
  set _base6;

  if count=. then
    count=0;
  indc=1;
  output _cnp;
  indc=2;

  if _trt=1 then
    do;
      count=_trt1 - count;
    end;

  if _trt=2 then
    do;
      count=_trt2 - count;
    end;
  output _cnp;

  if indc=2 and count=0 then
    output _tmp_cnp;
run;

proc sort data=_cnp;
  by _byvar1 _cat _rowsrt _rwlabel _trt;
run;

proc sort nodupkey data=_tmp_cnp(keep=_byvar1 _cat _rowsrt _rwlabel _trt);
  by _byvar1 _cat _rowsrt _rwlabel _trt;
run;

*****
* Call proc freq procedure to calculate CI for observed proportion  *;

```

\*\*\*\*\*,  
;

```
proc freq data=_cnp noprint;
  by _byvar1 _cat _rowsrt _rwlabel _trt;
  table indc/binomial alpha=0.05;
  output out=obsprop binomial;
  weight count;
run;

data obsprop;
  merge obsprop _tmp_cnp(in=a);
  by _byvar1 _cat _rowsrt _rwlabel _trt;

  if _bin_=1 and not a then
    do;
      xl_bin_=1 - xu_bin;
      xu_bin_=1 - xl_bin;
    end;
  else
    do;
      xl_bin_=xl_bin;
      xu_bin_=xu_bin;
    end;
run;

data cnpobsprop1(keep=_byvar1 _cat _rowsrt _rwlabel _trt cnp_ci);
  set obsprop;
  by _byvar1 _cat _rowsrt _rwlabel _trt;
  cnp_ci=(' || compress(put(xl_bin_ * 100, 5.1))
    || '(*ESC*){nbspace 1}' || compress(put(xu_bin_ * 100, 5.1)) || ');
  label cnp_ci='95% CI';
run;

proc datasets lib=work nolist gennum=all;
  delete _cnp obsprop;
run;

proc sort data=_base6;
  by _byvar1 _cat _rowsrt _rwlabel _trt;
run;

proc sort data=cnpobsprop1;
  by _byvar1 _cat _rowsrt _rwlabel _trt;
run;

data _base6;
  merge _base6(in=a) cnpobsprop1;
  by _byvar1 _cat _rowsrt _rwlabel _trt;

  if a;

  if compress(_cvalue)="0" then
    do;
      cnp_ci="NE";
```

FDA-CBER-2022-5812-0735742

```

end;

run;

*****
* Specification 5.4: Statistics for Chills category          *;
*****
* Specification 5.4.1: Count denominator (N)                *;
*****

data _anal7;
  set _anal1;
  _blcksrt=4;
run;

proc sort data=_anal7;
  by _datasrt _byvar1 _blcksrt DENOMFL _trt _cat;
run;

proc sort data=_anal7 out=_catby7(keep=_byvar1) nodupkey;
  by _byvar1;
  where paramcd eq upcase("Maxchil");
;
run;

data _temp7;
  set _anal7;
  output;
run;

proc sort data=_temp7 out=_temp97 nodupkey;
  by _datasrt _byvar1 _blcksrt _cat DENOMFL _trt usubjid;
  where paramcd eq upcase("Maxchil");
run;

proc freq data=_temp97 noprint;
  format DENOMFL;
  tables _datasrt*_byvar1*_blcksrt*_cat * DENOMFL * _trt / sparse norow nocol
    nopercnt out=_pct7(drop=percent);
run;

proc freq data=_pct7 noprint;
  where DENOMFL ne 9999;
  weight count;
  tables _datasrt*_byvar1*_cat * _trt / sparse noprint out=_denom7(drop=percent);
run;

data _denomf7;
  length _cat $100;
  _datasrt=1;
  set _catby7(keep=_byvar1);
  * All treatment groups ;
  _trt1=0;
  _trt2=0;

```



```

* _CAT is the subgroup variable ;
_cat="12-15 Years ";
output;
_cat="16-25 Years ";
output;
run;

proc transpose data=_denom7 out=_denomin7(drop=_name __label_) prefix=_trt;
  by _datasrt _byvar1 _cat;
  var count;
  id _trt;
run;

data _frame7;
  set _frame3;
  _blcksrt=4;
run;

proc sort data=_frame7;
  by _datasrt _byvar1 _blcksrt _cat DENOMFL _trt;
run;

proc sort data=_pct7;
  by _datasrt _byvar1 _blcksrt _cat DENOMFL _trt;
run;

data _pct7;
  merge _frame7(in=_inframe) _pct7;
  by _datasrt _byvar1 _blcksrt _cat DENOMFL _trt;

  if _inframe;

  if count=. then
    count=0;
run;

proc sort data=_pct7;
  by _datasrt _byvar1 _blcksrt DENOMFL;
run;

data _miss7(keep=_datasrt _byvar1 _blcksrt DENOMFL totcount);
  set _pct7;
  where DENOMFL=9998;
  retain totcount;
  by _datasrt _byvar1 _blcksrt DENOMFL;

  if first.DENOMFL then
    totcount=0;
  totcount=totcount+count;

  if last.DENOMFL;
run;

data _pct7(drop=totcount);

```

```

merge _pct7 _miss7;
by _datasrt _byvar1 _blcksrt DENOMFL;

if totcount=0 then
    delete;
run;

proc sort data=_denomf7;
    by _datasrt _byvar1 _cat;
run;

proc sort data=_denomin7;
    by _datasrt _byvar1 _cat;
run;

data _denomin7;
    merge _denomf7(in=_inframe) _denomin7;
    by _datasrt _byvar1 _cat;

    if _inframe;
        _blcksrt=4;
run;

proc sort data=_pct7;
    by _datasrt _byvar1 _cat;
run;

data _pct7;
    if 0 then
        set _basetemplate;
    merge _denomin7(in=_a) _pct7;
    by _datasrt _byvar1 _cat;

    if _a;
        _varname="DENOMFL ";
        _vrlabel="Chills(*ESC*){super d} ";
        _rwlabel=put(DENOMFL, sev.);

    if DENOMFL=9998 then
        do;
            _rwlabel="Missing ";
            _catord=9998;
        end;
    else if DENOMFL=9999 then
        do;
            _rwlabel="Total ";
            _catord=9999;
        end;

    if _catord=. then
        _catord=9997;
run;

proc sort data=_pct7;

```

```

by _datasrt _byvar1 _blcksrt _catord DENOMFL _trt _cat;
run;

data _base7;
length _catlabl $200;
set _pct7 end=eof;
by _datasrt _byvar1 _blcksrt _catord DENOMFL _trt _cat;
retain _rowsrt 0 _rowmax 0;
array _trtcnt(*) _trt1-_trt3;
drop _rowmax _cpct;
length _cpct $100;
_cpct=' ';
_module='mcatstat';

if count > . then
    _cvalue=put(count, 5.);
else
    _cvalue=put(0, 5.);

if length(_cvalue) < 5 then
    do;
        *-----;
        * Put character A0x at right most character to pad text;
        *-----;
        substr(_cvalue, 5, 1)='A0'x;
    end;

if first._byvar1 then
    _rowsrt=0;

if first.DENOMFL then
    do;
        _rowsrt=_rowsrt + 1;
        _rowmax=max(_rowsrt, _rowmax);
    end;
_datatyp='data';
_indent=0;
_dptindt=0;
_vorder=1;
_rowjump=1;

if upcase(_rwlabel)='_NONE_' then
    _rwlabel=' ';
_indent=8;
_dptindt=0;

if _trt=2 +1 then
    _trt=9999;

if eof then
    call symput('_rowsrt', compress(put(_rowmax, 4.)));
_direct="TOP ";
_p=0;
run;

```

```

*****
* Specification 5.4.2: Count n and percentage (%) for individual severity    *;
*****
;

data _anal8;
    set _anal2;
    _blcksrt=4;
run;

proc sort data=_anal8;
    by _datasrt _byvar1 _blcksrt AVAL _trt _cat;
run;

proc sort data=_anal8 out=_catby8(keep=_byvar1) nodupkey;
    by _byvar1;
    where paramcd eq upcase("Maxchil") and ex_none_flg=0 and eventfl="Y";
run;

data _temp8;
    set _anal8;
    output;
run;

proc sort data=_temp8 out=_temp98 nodupkey;
    by _datasrt _byvar1 _blcksrt _cat AVAL _trt usubjid;
    where paramcd eq upcase("Maxchil") and ex_none_flg=0 and eventfl="Y";
run;

proc freq data=_temp98 noprint;
    format AVAL;
    tables _datasrt*_byvar1*_blcksrt*_cat * AVAL * _trt / sparse norow nocol
        nopercnt out=_pct8(drop=percent);
run;

proc sort data=_temp8 out=_analcnt8 nodupkey;
    by _datasrt _byvar1 _cat _trt USUBJID;
    where paramcd eq upcase("Maxchil") and knowvfl eq 'Y';
run;

proc freq data=_analcnt8 noprint;
    tables _datasrt*_byvar1*_cat * _trt / sparse noprint out=_denom8(drop=percent);
run;

data _denomf8;
    length _cat $100;
    _datasrt=1;
    set _catby8(keep=_byvar1);
    * All treatment groups ;
    _trt1=0;
    _trt2=0;
    * _CAT is the subgroup variable ;
    _cat="12-15 Years ";
    output;

```

```

    _cat="16-25 Years ";
    output;
run;

proc transpose data=_denom8 out=_denomin8(drop=_name_ _label_) prefix=_trt;
    by _datasrt _byvar1 _cat;
    var count;
    id _trt;
run;

data _frame8;
    set _frame4;
    _blcksrt=4;
run;

proc sort data=_frame8;
    by _datasrt _byvar1 _blcksrt _cat AVAL _trt;
run;

proc sort data=_pct8;
    by _datasrt _byvar1 _blcksrt _cat AVAL _trt;
run;

data _pct8;
    merge _frame8(in=_inframe) _pct8;
    by _datasrt _byvar1 _blcksrt _cat AVAL _trt;

    if _inframe;

    if count=. then
        count=0;
run;

proc sort data=_pct8;
    by _datasrt _byvar1 _blcksrt AVAL;
run;

data _miss8(keep=_datasrt _byvar1 _blcksrt AVAL totcount);
    set _pct8;
    where AVAL=9998;
    retain totcount;
    by _datasrt _byvar1 _blcksrt AVAL;

    if first.AVAL then
        totcount=0;
    totcount=totcount+count;

    if last.AVAL;
run;

data _pct8(drop=totcount);
    merge _pct8 _miss8;
    by _datasrt _byvar1 _blcksrt AVAL;

```

```

        if totcount=0 then
            delete;
run;

proc sort data=_denomf8;
    by _datasrt _byvar1 _cat;
run;

proc sort data=_denomin8;
    by _datasrt _byvar1 _cat;
run;

data _denomin8;
    merge _denomf8(in=_inframe) _denomin8;
    by _datasrt _byvar1 _cat;

    if _inframe;
        _blcksrt=4;
run;

proc sort data=_pct8;
    by _datasrt _byvar1 _cat;
run;

data _pct8;
    if 0 then
        set _basetemplate;
    merge _denomin8(in=_a) _pct8;
    by _datasrt _byvar1 _cat;

    if _a;
        _varname="AVAL ";
        _vrlabel=" ";
        _rwlabel=put(AVAL, sev.);

    if AVAL=9998 then
        do;
            _rwlabel="Missing ";
            _catord=9998;
        end;
    else if AVAL=9999 then
        do;
            _rwlabel="Total ";
            _catord=9999;
        end;

    if _catord=. then
        _catord=9997;
run;

proc sort data=_pct8;
    by _datasrt _byvar1 _blcksrt _catord AVAL _trt _cat;
run;

```

```

data _base8;
  length _catlabl $200;
  set _pct8 end=eof;
  by _datasrt _byvar1 _blcksrt _catord AVAL _trt _cat;
  retain _rowsrt 5 _rowmax 0;
  array _trtcnt(*) _trt1-_trt3;
  drop _rowmax _cpct;
  length _cpct $100;
  _cpct='';
  _module='mcatstat';

  if count > . then
    _cvalue=put(count, 5.);
  else
    _cvalue=put(0, 5.);
  *-----;
  * Format percent to append to display value in _CVALUE ;
  *-----;

  if _trt ne . then
    do;

      if _trtcnt(_trt) > 0 then
        do;
          percent=count / _trtcnt(_trt) * 100;

          if percent > 0 then
            do;

              if round(percent, 0.1) GE 0.1 then
                _cpct="(*ESC*){nbspspace 1}("||strip(put(percent, 5.1))||")";
              else
                _cpct="(*ESC*){nbspspace 1}(0.0)";
              _cvalue=trim(_cvalue)||_cpct;
            end;
          end;
        end;
      end;

    end;

  if length(_cvalue) < 13 then
    do;
      *-----;
      * Put character A0x at right most character to pad text;
      *-----;
      substr(_cvalue, 13, 1)='A0'x;
    end;

  if first._byvar1 then
    _rowsrt=6;

  if first.AVAL then
    do;
      _rowsrt=_rowsrt + 1;
      _rowmax=max(_rowsrt, _rowmax);
    end;

```

```

    _datatype='data';
    _indent=0;
    _dptindt=0;
    _vorder=1;
    _rowjump=1;

    if upcase(_rwlabel)='_NONE_' then
        _rwlabel=' ';
        _indent=8;
        _dptindt=0;

    if _trt=2 +1 then
        _trt=9999;

    if eof then
        call symput('_rowsrt', compress(put(_rowmax, 4.)));
        _direct="TOP ";
        _p=2;
run;

```

```

data _base8;
    set _base8;
    length _cvalue2 $30.;
    _cvalue2=strip(tranwrd(_cvalue, 'A0'x, ""));
    _cvalue21=strip(scan(_cvalue, 1, '('));
    _cvalue22=compress(scan(_cvalue, 2, '(', ')'));
run;

```

```

data _base7;
    set _base7;
    drop _trt1 _trt2 count;
run;

```

```

proc sort data=_base7;
    by _datasrt _byvar1 _cat _trt;
run;

```

```

data _base7;
    set _base7;

    if _trt=1 then
        do;
            _trt1=input(_cvalue, ??best.);
        end;

    if _trt=2 then
        do;
            _trt2=input(_cvalue, ??best.);
        end;
run;

```

```

proc sort data=_base8(keep=_datasrt _trt _cvalue2 _cvalue21 _cvalue22 _cat
    _byvar1 count);

```



```

by _datasrt _byvar1 _cat _trt;
quit;

data _base8;
merge _base7(in=a) _base8(in=b);
by _datasrt _byvar1 _cat _trt;

if a;

if a and not b then
do;
    _cvalue2="0";
    _cvalue21="0";
end;

if compress(_cvalue2)="0" then
    _cvalue22=put(0, 5.1);

if compress(_cvalue)="0" then
do;
    _cvalue2="NA";
    _cvalue21="NA";
    _cvalue22="NA";
end;

run;

data _base7;
set _base7;
delete;

run;

*****
* Specification 5.4.3: Calculate 95% CI for observed proportion          *;
*****

data _cnp _tmp_cnp;
set _base8;

if count=. then
count=0;
indc=1;
output _cnp;
indc=2;

if _trt=1 then
do;
count=_trt1 - count;
end;

if _trt=2 then
do;
count=_trt2 - count;
end;
output _cnp;

```

```

        if indc=2 and count=0 then
            output _tmp_cnp;
run;

proc sort data=_cnp;
    by _byvar1 _cat _rowsrt _rwlabel _trt;
run;

proc sort nodupkey data=_tmp_cnp(keep=_byvar1 _cat _rowsrt _rwlabel _trt);
    by _byvar1 _cat _rowsrt _rwlabel _trt;
run;

*****
* Call proc freq procedure to calculate CI for observed proportion      *;
*****;

proc freq data=_cnp noprint;
    by _byvar1 _cat _rowsrt _rwlabel _trt;
    table indc/binomial alpha=0.05;
    output out=obsprop binomial;
    weight count;
run;

data obsprop;
    merge obsprop _tmp_cnp(in=a);
    by _byvar1 _cat _rowsrt _rwlabel _trt;

    if _bin_=1 and not a then
        do;
            xl_bin_=1 - xu_bin;
            xu_bin_=1 - xl_bin;
        end;
    else
        do;
            xl_bin_=xl_bin;
            xu_bin_=xu_bin;
        end;
run;

data cnpobsprop1(keep=_byvar1 _cat _rowsrt _rwlabel _trt cnp_ci);
    set obsprop;
    by _byvar1 _cat _rowsrt _rwlabel _trt;
    cnp_ci=(' || compress(put(xl_bin_ * 100, 5.1))
            || '(*ESC*){nbspspace 1}' || compress(put(xu_bin_ * 100, 5.1)) || ');
    label cnp_ci='95% CI';
run;

proc datasets lib=work nolist gennum=all;
    delete _cnp obsprop;
run;

proc sort data=_base8;
    by _byvar1 _cat _rowsrt _rwlabel _trt;
run;

```

```

proc sort data=cnpobsprop1;
  by _byvar1 _cat _rowsrt _rwlabel _trt;
run;

data _base8;
  merge _base8(in=a) cnpobsprop1;
  by _byvar1 _cat _rowsrt _rwlabel _trt;

  if a;

  if compress(_cvalue)="0" then
    do;
      cnp_ci="NE";
    end;
run;

*****
* Specification 5.5: Statistics for Vomiting category *;
*****
* Specification 5.5.1: Count denominator (N) *;
*****

data _anal9;
  set _anal1;
  _blcksrt=5;
run;

proc sort data=_anal9;
  by _datasrt _byvar1 _blcksrt DENOMFL _trt _cat;
run;

proc sort data=_anal9 out=_catby9(keep=_byvar1) nodupkey;
  by _byvar1;
  where paramcd eq upcase("Maxsvom");
run;

data _temp9;
  set _anal9;
  output;
run;

proc sort data=_temp9 out=_temp99 nodupkey;
  by _datasrt _byvar1 _blcksrt _cat DENOMFL _trt usubjid;
  where paramcd eq upcase("Maxsvom");
run;

proc freq data=_temp99 noprint;
  format DENOMFL;
  tables _datasrt*_byvar1*_blcksrt*_cat * DENOMFL * _trt / sparse norow nocol
    nopercnt out=_pct9(drop=percent);
run;

proc freq data=_pct9 noprint;

```

```

where DENOMFL ne 9999;
weight count;
tables _datasrt*_byvar1*_cat *_trt / sparse noprint out=_denom9(drop=percent);
run;

data _denomf9;
length _cat $100;
_datasrt=1;
set _catby9(keep=_byvar1);
* All treatment groups ;
_trt1=0;
_trt2=0;
* _CAT is the subgroup variable ;
_cat="12-15 Years ";
output;
_cat="16-25 Years ";
output;
run;

proc transpose data=_denom9 out=_denomin9(drop=_name__label_) prefix=_trt;
by _datasrt _byvar1 _cat;
var count;
id _trt;
run;

data _frame9;
set _frame3;
_blcksrt=5;
run;

proc sort data=_frame9;
by _datasrt _byvar1 _blcksrt _cat DENOMFL _trt;
run;

proc sort data=_pct9;
by _datasrt _byvar1 _blcksrt _cat DENOMFL _trt;
run;

data _pct9;
merge _frame9(in=_inframe) _pct9;
by _datasrt _byvar1 _blcksrt _cat DENOMFL _trt;

if _inframe;

if count=. then
count=0;
run;

proc sort data=_pct9;
by _datasrt _byvar1 _blcksrt DENOMFL;
run;

data _miss9(keep=_datasrt _byvar1 _blcksrt DENOMFL totcount);

```

```

set _pct9;
where DENOMFL=9998;
retain totcount;
by _datasrt _byvar1 _blcksrt DENOMFL;

if first.DENOMFL then
    totcount=0;
totcount=totcount+count;

if last.DENOMFL;
run;

data _pct9(drop=totcount);
merge _pct9 _miss9;
by _datasrt _byvar1 _blcksrt DENOMFL;

if totcount=0 then
    delete;
run;

proc sort data=_denomf9;
    by _datasrt _byvar1 _cat;
run;

proc sort data=_denomin9;
    by _datasrt _byvar1 _cat;
run;

data _denomin9;
merge _denomf9(in=_inframe) _denomin9;
by _datasrt _byvar1 _cat;

if _inframe;
    _blcksrt=5;
run;

proc sort data=_pct9;
    by _datasrt _byvar1 _cat;
run;

data _pct9;
if 0 then
    set _basetemplate;
merge _denomin9(in=_a) _pct9;
by _datasrt _byvar1 _cat;

if _a;
    _varname="DENOMFL ";
    _vrlabel="Vomiting(*ESC*){super e} ";
    _rwlabel=put(DENOMFL, sev.);

if DENOMFL=9998 then
    do;
        _rwlabel="Missing ";

```

```

        _catord=9998;
    end;
else if DENOMFL=9999 then
    do;
        _rwlabel="Total ";
        _catord=9999;
    end;

    if _catord=. then
        _catord=9997;

run;

proc sort data=_pct9;
    by _datasrt _byvar1 _blcksrt _catord DENOMFL _trt _cat;
run;

data _base9;
    length _catlabl $200;
    set _pct9 end=eof;
    by _datasrt _byvar1 _blcksrt _catord DENOMFL _trt _cat;
    retain _rowsrt 0 _rowmax 0;
    array _trtcnt(*) _trt1-_trt3;
    drop _rowmax _cpct;
    length _cpct $100;
    _cpct='';
    _module='mcatstat';

    if count > . then
        _cvalue=put(count, 5.);
    else
        _cvalue=put(0, 5.);

    if length(_cvalue) < 5 then
        do;
            *-----;
            * Put character A0x at right most character to pad text;
            *-----;
            substr(_cvalue, 5, 1)='A0'x;
        end;

    if first._byvar1 then
        _rowsrt=0;

    if first.DENOMFL then
        do;
            _rowsrt=_rowsrt + 1;
            _rowmax=max(_rowsrt, _rowmax);
        end;
    _datatyp='data';
    _indent=0;
    _dptindt=0;
    _vorder=1;
    _rowjump=1;

```

```

if upcase(_rwlabel)='_NONE_' then
  _rwlabel=' ';
  _indent=8;
  _dptindt=0;

if _trt=2 +1 then
  _trt=9999;

if eof then
  call symput('_rowsrt', compress(put(_rowmax, 4.)));
  _direct="TOP ";
  _p=0;
run;

*****
* Specification 5.5.2: Count n and percentage (%) for individual severity    *;
*****

data _anal10;
  set _anal2;
  _blcksrt=5;
  _cnt=1;
run;

proc sort data=_anal10;
  by _datasrt _byvar1 _blcksrt AVAL _trt _cat;
run;

proc sort data=_anal10 out=_catby10(keep=_byvar1) nodupkey;
  by _byvar1;
  where paramcd eq upcase("Maxsvom") and ex_none_flg=0 and eventfl="Y";
run;

data _temp10;
  set _anal10;
  output;
run;

proc sort data=_temp10 out=_temp910 nodupkey;
  by _datasrt _byvar1 _blcksrt _cat AVAL _trt usubjid;
  where paramcd eq upcase("Maxsvom") and ex_none_flg=0 and eventfl="Y";
run;

proc freq data=_temp910 noprint;
  format AVAL;
  tables _datasrt*_byvar1*_blcksrt*_cat * AVAL * _trt / sparse norow nocol
    nopercnt out=_pct10(drop=percent);
run;

proc sort data=_temp10 out=_analcnt10 nodupkey;
  by _datasrt _byvar1 _cat _trt USUBJID;
  where paramcd eq upcase("Maxsvom") and knowvfl eq 'Y';
run;

```

```

proc freq data=_analcnt10 noprint;
  tables _datasrt*_byvar1*_cat * _trt / sparse noprint
    out=_denom10(drop=percent);
run;

data _denomf10;
  length _cat $100;
  _datasrt=1;
  set _catby10(keep=_byvar1);
  * All treatment groups ;
  _trt1=0;
  _trt2=0;
  * _CAT is the subgroup variable ;
  _cat="12-15 Years ";
  output;
  _cat="16-25 Years ";
  output;
run;

proc transpose data=_denom10 out=_denomin10(drop=_name__label_) prefix=_trt;
  by _datasrt _byvar1 _cat;
  var count;
  id _trt;
run;
data _frame10;
  set _frame4;
  _blcksrt=5;
run;

proc sort data=_frame10;
  by _datasrt _byvar1 _blcksrt _cat AVAL _trt;
run;

proc sort data=_pct10;
  by _datasrt _byvar1 _blcksrt _cat AVAL _trt;
run;

data _pct10;
  merge _frame10(in=_inframe) _pct10;
  by _datasrt _byvar1 _blcksrt _cat AVAL _trt;

  if _inframe;

  if count=. then
    count=0;
run;

proc sort data=_pct10;
  by _datasrt _byvar1 _blcksrt AVAL;
run;

data _miss10(keep=_datasrt _byvar1 _blcksrt AVAL totcount);
  set _pct10;

```



```

where AVAL=9998;
retain totcount;
by _datasrt _byvar1 _blcksrt AVAL;

if first.AVAL then
    totcount=0;
totcount=totcount+count;

if last.AVAL;
run;

data _pct10(drop=totcount);
merge _pct10 _miss10;
by _datasrt _byvar1 _blcksrt AVAL;

if totcount=0 then
    delete;
run;

proc sort data=_denomf10;
by _datasrt _byvar1 _cat;
run;

proc sort data=_denomin10;
by _datasrt _byvar1 _cat;
run;

data _denomin10;
merge _denomf10(in=_inframe) _denomin10;
by _datasrt _byvar1 _cat;

if _inframe;
    _blcksrt=5;
run;

proc sort data=_pct10;
by _datasrt _byvar1 _cat;
run;

data _pct10;
if 0 then
    set _basetemplate;
merge _denomin10(in=_a) _pct10;
by _datasrt _byvar1 _cat;

if _a;
    _varname="AVAL ";
    _vrlabel=" ";
    _rwlabel=put(AVAL, sev.);

if AVAL=9998 then
    do;
        _rwlabel="Missing ";
        _catord=9998;

```

```

        end;
    else if AVAL=9999 then
        do;
            _rwlabel="Total ";
            _catord=9999;
        end;

    if _catord=. then
        _catord=9997;

run;

proc sort data=_pct10;
    by _datasrt _byvar1 _blcksrt _catord AVAL _trt _cat;
run;

data _base10;
    length _catlabl $200;
    set _pct10 end=eof;
    by _datasrt _byvar1 _blcksrt _catord AVAL _trt _cat;
    retain _rowsrt 5 _rowmax 0;
    array _trtcnt(*) _trt1-_trt3;
    drop _rowmax _cpct;
    length _cpct $100;
    _cpct=' ';
    _module='mcatstat';

    if count > . then
        _cvalue=put(count, 5.);
    else
        _cvalue=put(0, 5.);
    *-----;
    * Format percent to append to display value in _CVALUE ;
    *-----;

    if _trt ne . then
        do;

            if _trtcnt(_trt) > 0 then
                do;
                    percent=count / _trtcnt(_trt) * 100;

                    if percent > 0 then
                        do;

                            if round(percent, 0.1) GE 0.1 then
                                _cpct="(*ESC*){nbspspace 1}("||strip(put(percent, 5.1))||")";
                            else
                                _cpct="(*ESC*){nbspspace 1}{(0.0)}";
                            _cvalue=trim(_cvalue)||_cpct;
                        end;
                    end;
                end;
            end;

        end;

    if length(_cvalue) < 13 then

```

```

do;
    *-----;
    * Put character A0x at right most character to pad text;
    *-----;
    substr(_cvalue, 13, 1)='A0'x;
end;

if first._byvar1 then
    _rowsrt=6;

if first.AVAL then
do;
    _rowsrt=_rowsrt + 1;
    _rowmax=max(_rowsrt, _rowmax);
end;
_datatyp='data';
_indent=0;
_dptindt=0;
_vorder=1;
_rowjump=1;

if upcase(_rwlabel)='_NONE_' then
    _rwlabel=' ';
_indent=8;
_dptindt=0;

if _trt=2 +1 then
    _trt=9999;

if eof then
    call symput('_rowsrt', compress(put(_rowmax, 4.)));
_direct="TOP ";
_p=2;
run;

data _base10;
set _base10;
length _cvalue2 $30.;
_cvalue2=strip(tranwrd(_cvalue, 'A0'x, ""));
_cvalue21=strip(scan(_cvalue, 1, '('));
_cvalue22=compress(scan(_cvalue, 2, '(', ')'));
run;

data _base9;
set _base9;
drop _trt1 _trt2 count;
run;

proc sort data=_base9;
by _datasrt _byvar1 _cat _trt;
run;

data _base9;
set _base9;

```

```

if _trt=1 then
  do;
    _trt1=input(_cvalue, ??best.);
  end;

if _trt=2 then
  do;
    _trt2=input(_cvalue, ??best.);
  end;

run;

proc sort data=_base10(keep=_datasrt _trt _cvalue2 _cvalue21 _cvalue22 _cat
  _byvar1 count);
  by _datasrt _byvar1 _cat _trt;
quit;

data _base10;
  merge _base9(in=a) _base10(in=b);
  by _datasrt _byvar1 _cat _trt;

  if a;

  if a and not b then
    do;
      _cvalue2="0";
      _cvalue21="0";
    end;

  if compress(_cvalue2)="0" then
    _cvalue22=put(0, 5.1);

  if compress(_cvalue)="0" then
    do;
      _cvalue2="NA";
      _cvalue21="NA";
      _cvalue22="NA";
    end;

run;

data _base9;
  set _base9;
  delete;

run;

```

```

*****
* Specification 5.5.3: Calculate 95% CI for observed proportion      *;
*****

```

```

data _cnp _tmp_cnp;
  set _base10;

  if count=. then
    count=0;

```

```

indc=1;
output _cnp;
indc=2;

if _trt=1 then
  do;
    count=_trt1 - count;
  end;

if _trt=2 then
  do;
    count=_trt2 - count;
  end;
output _cnp;

if indc=2 and count=0 then
  output _tmp_cnp;
run;

proc sort data=_cnp;
  by _byvar1 _cat _rowsrt _rwlabel _trt;
run;

proc sort nodupkey data=_tmp_cnp(keep=_byvar1 _cat _rowsrt _rwlabel _trt);
  by _byvar1 _cat _rowsrt _rwlabel _trt;
run;

*****
* Call proc freq procedure to calculate CI for observed proportion          *;
*****

proc freq data=_cnp noprint;
  by _byvar1 _cat _rowsrt _rwlabel _trt;
  table indc/binomial alpha=0.05;
  output out=obsprop binomial;
  weight count;
run;

data obsprop;
  merge obsprop _tmp_cnp(in=a);
  by _byvar1 _cat _rowsrt _rwlabel _trt;

  if _bin_=1 and not a then
    do;
      xl_bin_=1 - xu_bin;
      xu_bin_=1 - xl_bin;
    end;
  else
    do;
      xl_bin_=xl_bin;
      xu_bin_=xu_bin;
    end;
run;

```

```

data cnpobsprop1(keep=_byvar1 _cat _rowsrt _rwlabel _trt cnp_ci);
  set obsprop;
  by _byvar1 _cat _rowsrt _rwlabel _trt;
  cnp_ci=(' || compress(put(xl_bin_ * 100, 5.1))
    || '(*ESC*){nbspace 1}' || compress(put(xu_bin_ * 100, 5.1)) || ');
  label cnp_ci='95% CI';
run;

```

```

proc datasets lib=work nolist gennum=all;
  delete _cnp obsprop;
run;

```

```

proc sort data=_base10;
  by _byvar1 _cat _rowsrt _rwlabel _trt;
run;

```

```

proc sort data=cnpobsprop1;
  by _byvar1 _cat _rowsrt _rwlabel _trt;
run;

```

```

data _base10;
  merge _base10(in=a) cnpobsprop1;
  by _byvar1 _cat _rowsrt _rwlabel _trt;

  if a;

  if compress(_cvalue)="0" then
    do;
      cnp_ci="NE";
    end;
run;

```

```

*****
* Specification 5.6: Statistics for Diarrhea category *;
*****
* Specification 5.6.1: Count denominator (N) *;
*****

```

```

data _anal11;
  set _anal1;
  _blcksrt=6;
run;

```

```

proc sort data=_anal11;
  by _datasrt _byvar1 _blcksrt DENOMFL _trt _cat;
run;

```

```

proc sort data=_anal11 out=_catby11(keep=_byvar1) nodupkey;
  by _byvar1;
  where paramcd eq upcase("Maxdiar");
run;

```

```

data _temp11;
    set _anall1;
    output;
run;

proc sort data=_temp11 out=_temp911 nodupkey;
    by _datasrt _byvar1 _blcksrt _cat DENOMFL _trt usubjid;
    where paramcd eq upcase("Maxdiar");
run;

proc freq data=_temp911 noprint;
    format DENOMFL;
    tables _datasrt*_byvar1*_blcksrt*_cat * DENOMFL * _trt / sparse norow nocol
        nopercnt out=_pct11(drop=percent);
run;

proc freq data=_pct11 noprint;
    where DENOMFL ne 9999;
    weight count;
    tables _datasrt*_byvar1*_cat * _trt / sparse noprint
        out=_denom11(drop=percent);
run;

data _denomf11;
    length _cat $100;
    _datasrt=1;
    set _catby11(keep=_byvar1);
    * All treatment groups ;
    _trt1=0;
    _trt2=0;
    * _CAT is the subgroup variable ;
    _cat="12-15 Years ";
    output;
    _cat="16-25 Years ";
    output;
run;

proc transpose data=_denom11 out=_denomin11(drop=_name__label_) prefix=_trt;
    by _datasrt _byvar1 _cat;
    var count;
    id _trt;
run;

data _frame11;
    set _frame3;
    _blcksrt=6;
run;

proc sort data=_frame11;
    by _datasrt _byvar1 _blcksrt _cat DENOMFL _trt;
run;

proc sort data=_pct11;
    by _datasrt _byvar1 _blcksrt _cat DENOMFL _trt;

```

```

run;

data _pct11;
  merge _frame11(in=_inframe) _pct11;
  by _datasrt _byvar1 _blcksrt _cat DENOMFL _trt;

  if _inframe;

  if count=. then
    count=0;
run;

proc sort data=_pct11;
  by _datasrt _byvar1 _blcksrt DENOMFL;
run;

data _miss11(keep=_datasrt _byvar1 _blcksrt DENOMFL totcount);
  set _pct11;
  where DENOMFL=9998;
  retain totcount;
  by _datasrt _byvar1 _blcksrt DENOMFL;

  if first.DENOMFL then
    totcount=0;
  totcount=totcount+count;

  if last.DENOMFL;
run;

data _pct11(drop=totcount);
  merge _pct11 _miss11;
  by _datasrt _byvar1 _blcksrt DENOMFL;

  if totcount=0 then
    delete;
run;

proc sort data=_denomf11;
  by _datasrt _byvar1 _cat;
run;

proc sort data=_denomin11;
  by _datasrt _byvar1 _cat;
run;

data _denomin11;
  merge _denomf11(in=_inframe) _denomin11;
  by _datasrt _byvar1 _cat;

  if _inframe;
  _blcksrt=6;
run;

proc sort data=_pct11;

```



```

    by _datasrt _byvar1 _cat;
run;

data _pct11;
    if 0 then
        set _basetemplate;
    merge _denomin11(in=_a) _pct11;
    by _datasrt _byvar1 _cat;

    if _a;
    _varname="DENOMFL ";
    _vrlabel="Diarrhea(*ESC*){super f} ";
    _rwlabel=put(DENOMFL, sev.);

    if DENOMFL=9998 then
        do;
            _rwlabel="Missing ";
            _catord=9998;
        end;
    else if DENOMFL=9999 then
        do;
            _rwlabel="Total ";
            _catord=9999;
        end;

    if _catord=. then
        _catord=9997;
run;

proc sort data=_pct11;
    by _datasrt _byvar1 _blcksrt _catord DENOMFL _trt _cat;
run;

data _base11;
    length _catlab1 $200;
    set _pct11 end=eof;
    by _datasrt _byvar1 _blcksrt _catord DENOMFL _trt _cat;
    retain _rowsrt 0 _rowmax 0;
    array _trtcnt(*) _trt1-_trt3;
    drop _rowmax _cpct;
    length _cpct $100;
    _cpct=' ';
    _module='mcatstat';

    if count > . then
        _cvalue=put(count, 5.);
    else
        _cvalue=put(0, 5.);

    if length(_cvalue) < 5 then
        do;
            *-----;
            * Put character A0x at right most character to pad text;
            *-----;

```

```

        substr(_cvalue, 5, 1)='A0'x;
    end;

    if first._byvar1 then
        _rowsrt=0;

    if first.DENOMFL then
        do;
            _rowsrt=_rowsrt + 1;
            _rowmax=max(_rowsrt, _rowmax);
        end;
        _datatyp='data';
        _indent=0;
        _dptindt=0;
        _vorder=1;
        _rowjump=1;

    if upcase(_rwlabel)='_NONE_' then
        _rwlabel=' ';
        _indent=8;
        _dptindt=0;

    if _trt=2 +1 then
        _trt=9999;

    if eof then
        call symput('_rowsrt', compress(put(_rowmax, 4.)));
        _direct="TOP ";
        _p=0;
run;

*****
* Specification 5.6.2: Count n and percentage (%) for individual severity    *;
*****

data _anal12;
    set _anal2;
    _blcksrt=6;

run;

proc sort data=_anal12;
    by _datasrt _byvar1 _blcksrt AVAL _trt _cat;
run;

proc sort data=_anal12 out=_catby12(keep=_byvar1) nodupkey;
    by _byvar1;
    where paramcd eq upcase("Maxdiar") and ex_none_flg=0 and eventfl="Y";
run;

data _temp12;
    set _anal12;
    output;
run;

```

```

proc sort data=_temp12 out=_temp912 nodupkey;
  by _datasrt _byvar1 _blcksrt _cat AVAL _trt usubjid;
  where paramcd eq upcase("Maxdiar") and ex_none_flg=0 and eventfl="Y";
run;

proc freq data=_temp912 noprint;
  format AVAL;
  tables _datasrt*_byvar1*_blcksrt*_cat * AVAL * _trt / sparse norow nocol
    nopercnt out=_pct12(drop=percent);
run;

proc sort data=_temp12 out=_analcnt12 nodupkey;
  by _datasrt _byvar1 _cat _trt USUBJID;
  where paramcd eq upcase("Maxdiar") and knowvfl eq 'Y';
run;

proc freq data=_analcnt12 noprint;
  tables _datasrt*_byvar1*_cat * _trt / sparse noprint
    out=_denom12(drop=percent);
run;

data _denomf12;
  length _cat $100;
  _datasrt=1;
  set _catby12(keep=_byvar1);
  * All treatment groups ;
  _trt1=0;
  _trt2=0;
  * _CAT is the subgroup variable ;
  _cat="12-15 Years ";
  output;
  _cat="16-25 Years ";
  output;
run;

proc transpose data=_denom12 out=_denomin12(drop=_name __label_) prefix=_trt;
  by _datasrt _byvar1 _cat;
  var count;
  id _trt;
run;

data _frame12;
  set _frame4;
  _blcksrt=6;
run;

proc sort data=_frame12;
  by _datasrt _byvar1 _blcksrt _cat AVAL _trt;
run;

proc sort data=_pct12;
  by _datasrt _byvar1 _blcksrt _cat AVAL _trt;
run;

```

```

data _pct12;
  merge _frame12(in=_inframe) _pct12;
  by _datasrt _byvar1 _blcksrt _cat AVAL _trt;

  if _inframe;

  if count=. then
    count=0;
run;

proc sort data=_pct12;
  by _datasrt _byvar1 _blcksrt AVAL;
run;

data _miss12(keep=_datasrt _byvar1 _blcksrt AVAL totcount);
  set _pct12;
  where AVAL=9998;
  retain totcount;
  by _datasrt _byvar1 _blcksrt AVAL;

  if first.AVAL then
    totcount=0;
  totcount=totcount+count;

  if last.AVAL;
run;

data _pct12(drop=totcount);
  merge _pct12 _miss12;
  by _datasrt _byvar1 _blcksrt AVAL;

  if totcount=0 then
    delete;
run;

proc sort data=_denomf12;
  by _datasrt _byvar1 _cat;
run;

proc sort data=_denomin12;
  by _datasrt _byvar1 _cat;
run;

data _denomin12;
  merge _denomf12(in=_inframe) _denomin12;
  by _datasrt _byvar1 _cat;

  if _inframe;
  _blcksrt=6;
run;

proc sort data=_pct12;
  by _datasrt _byvar1 _cat;

```

```

run;

data _pct12;
  if 0 then
    set _basetemplate;
  merge _denomin12(in=_a) _pct12;
  by _datasrt _byvar1 _cat;

  if _a;
  _varname="AVAL ";
  _vrlabel=" ";
  _rwlabel=put(AVAL, sev.);

  if AVAL=9998 then
    do;
      _rwlabel="Missing ";
      _catord=9998;
    end;
  else if AVAL=9999 then
    do;
      _rwlabel="Total ";
      _catord=9999;
    end;

  if _catord=. then
    _catord=9997;

run;

proc sort data=_pct12;
  by _datasrt _byvar1 _blcksrt _catord AVAL _trt _cat;
run;

data _base12;
  length _catlabl $200;
  set _pct12 end=eof;
  by _datasrt _byvar1 _blcksrt _catord AVAL _trt _cat;
  retain _rowsrt 5 _rowmax 0;
  array _trtcnt(*) _trt1-_trt3;
  drop _rowmax _cpct;
  length _cpct $100;
  _cpct='';
  _module='mcatstat';

  if count > . then
    _cvalue=put(count, 5.);
  else
    _cvalue=put(0, 5.);
  *-----;
  * Format percent to append to display value in _CVALUE ;
  *-----;

  if _trt ne . then
    do;

```

```

        if _trtcnt(_trt) > 0 then
            do;
                percent=count / _trtcnt(_trt) * 100;

                if percent > 0 then
                    do;

                        if round(percent, 0.1) GE 0.1 then
                            _cpct="(*ESC*){nbspspace 1}("||strip(put(percent, 5.1))||")";
                        else
                            _cpct="(*ESC*){nbspspace 1}(0.0)";
                        _cvalue=trim(_cvalue)||_cpct;
                    end;
                end;
            end;

        end;

if length(_cvalue) < 13 then
    do;
        *-----;
        * Put character A0x at right most character to pad text;
        *-----;
        substr(_cvalue, 13, 1)='A0'x;
    end;

if first._byvar1 then
    _rowsrt=6;

if first.AVAL then
    do;
        _rowsrt=_rowsrt + 1;
        _rowmax=max(_rowsrt, _rowmax);
    end;
_datatyp='data';
_indent=0;
_dptindt=0;
_vorder=1;
_rowjump=1;

if upcase(_rwlabel)='_NONE_' then
    _rwlabel=' ';
_indent=8;
_dptindt=0;

if _trt=2 +1 then
    _trt=9999;

if eof then
    call symput('_rowsrt', compress(put(_rowmax, 4)));
_direct="TOP ";
_p=2;
run;

data _base12;
set _base12;

```

```

length _cvalue2 $30.;
_cvalue2=strip(tranwrd(_cvalue, 'A0'x, ""));
_cvalue21=strip(scan(_cvalue, 1, '('));
_cvalue22=compress(scan(_cvalue, 2, '(', ')'));
run;

data _base11;
set _base11;
drop _trt1 _trt2 count;
run;

proc sort data=_base11;
by _datasrt _byvar1 _cat _trt;
run;

data _base11;
set _base11;

if _trt=1 then
do;
_trt1=input(_cvalue, ??best.);
end;

if _trt=2 then
do;
_trt2=input(_cvalue, ??best.);
end;

run;

proc sort data=_base12(keep=_datasrt _trt _cvalue2 _cvalue21 _cvalue22 _cat
_byvar1 count);
by _datasrt _byvar1 _cat _trt;
quit;

data _base12;
merge _base11(in=a) _base12(in=b);
by _datasrt _byvar1 _cat _trt;

if a;

if a and not b then
do;
_cvalue2="0";
_cvalue21="0";
end;

if compress(_cvalue2)="0" then
_cvalue22=put(0, 5.1);

if compress(_cvalue)="0" then
do;
_cvalue2="NA";
_cvalue21="NA";
_cvalue22="NA";

```

```

        end;
run;

data _base11;
    set _base11;
    delete;
run;

*****
* Specification 5.6.3: Calculate 95% CI for observed proportion          *;
*****

data _cnp_tmp_cnp;
    set _base12;

    if count=. then
        count=0;
    indc=1;
    output _cnp;
    indc=2;

    if _trt=1 then
        do;
            count=_trt1 - count;
        end;

    if _trt=2 then
        do;
            count=_trt2 - count;
        end;
    output _cnp;

    if indc=2 and count=0 then
        output _tmp_cnp;
run;

proc sort data=_cnp;
    by _byvar1 _cat _rowsrt _rwlabel _trt;
run;

proc sort nodupkey data=_tmp_cnp(keep=_byvar1 _cat _rowsrt _rwlabel _trt);
    by _byvar1 _cat _rowsrt _rwlabel _trt;
run;

*****
* Call proc freq procedure to calculate CI for observed proportion      *;
*****

proc freq data=_cnp noprint;
    by _byvar1 _cat _rowsrt _rwlabel _trt;
    table indc/binomial alpha=0.05;
    output out=obsprop binomial;
    weight count;
run;

```



```

data obsprop;
  merge obsprop_tmp_cnp(in=a);
  by _byvar1_cat_rowsrt_rwlabel_trt;

  if _bin_=1 and not a then
    do;
      xl_bin_=1 - xu_bin;
      xu_bin_=1 - xl_bin;
    end;
  else
    do;
      xl_bin_=xl_bin;
      xu_bin_=xu_bin;
    end;
run;

data cnpobsprop1(keep=_byvar1_cat_rowsrt_rwlabel_trt_cnp_ci);
  set obsprop;
  by _byvar1_cat_rowsrt_rwlabel_trt;
  cnp_ci=(' || compress(put(xl_bin_ * 100, 5.1))
    || '(*ESC*){nbspace 1}' || compress(put(xu_bin_ * 100, 5.1)) || ');
  label cnp_ci='95% CI';
run;

proc datasets lib=work nolist gennum=all;
  delete _cnp obsprop;
run;

```

```

proc sort data=_base12;
  by _byvar1_cat_rowsrt_rwlabel_trt;
run;

```

```

proc sort data=cnpobsprop1;
  by _byvar1_cat_rowsrt_rwlabel_trt;
run;

```

```

data _base12;
  merge _base12(in=a) cnpobsprop1;
  by _byvar1_cat_rowsrt_rwlabel_trt;

  if a;

  if compress(_cvalue)="0" then
    do;
      cnp_ci="NE";
    end;
run;

```

```

*****
* Specification 5.7: Statistics for New or worsened muscle pain category      *;
*****
* Specification 5.7.1: Count denominator (N)                                *;
*****

```

```

data _anal13;
    set _anal1;
    _blcksrt=7;
run;

proc sort data=_anal13;
    by _datasrt _byvar1 _blcksrt DENOMFL _trt _cat;
run;

proc sort data=_anal13 out=_catby13(keep=_byvar1) nodupkey;
    by _byvar1;
    where paramcd eq upcase("Maxsmp");
run;

data _temp13;
    set _anal13;
    output;
run;

proc sort data=_temp13 out=_temp913 nodupkey;
    by _datasrt _byvar1 _blcksrt _cat DENOMFL _trt usubjid;
    where paramcd eq upcase("Maxsmp");
run;

proc freq data=_temp913 noprint;
    format DENOMFL;
    tables _datasrt*_byvar1*_blcksrt*_cat * DENOMFL * _trt / sparse norow nocol
        nopercnt out=_pct13(drop=percent);
run;

proc freq data=_pct13 noprint;
    where DENOMFL ne 9999;
    weight count;
    tables _datasrt*_byvar1*_cat * _trt / sparse noprint
        out=_denom13(drop=percent);
run;

data _denomf13;
    length _cat $100;
    _datasrt=1;
    set _catby13(keep=_byvar1);
    * All treatment groups ;
    _trt1=0;
    _trt2=0;
    * _CAT is the subgroup variable ;
    _cat="12-15 Years ";
    output;
    _cat="16-25 Years ";
    output;
run;

proc transpose data=_denom13 out=_denomin13(drop=_name _label_) prefix=_trt;
    by _datasrt _byvar1 _cat;
    var count;

```

```

        id_trt;
run;

data _frame13;
    set _frame3;
    _blcksrt=7;
run;

proc sort data=_frame13;
    by _datasrt _byvar1 _blcksrt _cat DENOMFL _trt;
run;

proc sort data=_pct13;
    by _datasrt _byvar1 _blcksrt _cat DENOMFL _trt;
run;

data _pct13;
    merge _frame13(in=_inframe) _pct13;
    by _datasrt _byvar1 _blcksrt _cat DENOMFL _trt;

    if _inframe;

    if count=. then
        count=0;
run;

proc sort data=_pct13;
    by _datasrt _byvar1 _blcksrt DENOMFL;
run;

data _miss13(keep=_datasrt _byvar1 _blcksrt DENOMFL totcount);
    set _pct13;
    where DENOMFL=9998;
    retain totcount;
    by _datasrt _byvar1 _blcksrt DENOMFL;

    if first.DENOMFL then
        totcount=0;
    totcount=totcount+count;

    if last.DENOMFL;
run;

data _pct13(drop=totcount);
    merge _pct13 _miss13;
    by _datasrt _byvar1 _blcksrt DENOMFL;

    if totcount=0 then
        delete;
run;

proc sort data=_denomf13;
    by _datasrt _byvar1 _cat;
run;

```

```

proc sort data=_denomin13;
    by _datasrt _byvar1 _cat;
run;

data _denomin13;
    merge _denomf13(in=_inframe) _denomin13;
    by _datasrt _byvar1 _cat;

    if _inframe;
        _blcksrt=7;
run;

proc sort data=_pct13;
    by _datasrt _byvar1 _cat;
run;

data _pct13;
    if 0 then
        set _basetemplate;
    merge _denomin13(in=_a) _pct13;
    by _datasrt _byvar1 _cat;

    if _a;
        _varname="DENOMFL ";
        _vrlabel="New or worsened muscle pain(*ESC*){super d} ";
        _rwlabel=put(DENOMFL, sev.);

    if DENOMFL=9998 then
        do;
            _rwlabel="Missing ";
            _catord=9998;
        end;
    else if DENOMFL=9999 then
        do;
            _rwlabel="Total ";
            _catord=9999;
        end;

    if _catord=. then
        _catord=9997;
run;

proc sort data=_pct13;
    by _datasrt _byvar1 _blcksrt _catord DENOMFL _trt _cat;
run;

data _base13;
    length _catlabl $200;
    set _pct13 end=eof;
    by _datasrt _byvar1 _blcksrt _catord DENOMFL _trt _cat;
    retain _rowsrt 0 _rowmax 0;
    array _trtcnt(*) _trt1-_trt3;
    drop _rowmax _cpct;

```

```

length _cpct $100;
_cpct='';
_module='mcatstat';

if count > . then
  _cvalue=put(count, 5.);
else
  _cvalue=put(0, 5.);

if length(_cvalue) < 5 then
  do;
    *-----;
    * Put character A0x at right most character to pad text;
    *-----;
    substr(_cvalue, 5, 1)='A0'x;
  end;

if first._byvar1 then
  _rowsrt=0;

if first.DENOMFL then
  do;
    _rowsrt=_rowsrt + 1;
    _rowmax=max(_rowsrt, _rowmax);
  end;
_datatyp='data';
_indent=0;
_dptindt=0;
_vorder=1;
_rowjump=1;

if upcase(_rwlabel)='_NONE_' then
  _rwlabel='';
_indent=8;
_dptindt=0;

if _trt=2 +1 then
  _trt=9999;

if eof then
  call symput('_rowsrt', compress(put(_rowmax, 4.)));
_direct="TOP ";
_p=0;

run;

*****
* Specification 5.7.2: Count n and percentage (%) for individual severity   *;
*****

data _anal14;
  set _anal2;
  _blcksrt=7;

run;

```

```

proc sort data=_anal14;
  by _datasrt _byvar1 _blcksrt AVAL _trt _cat;
run;

proc sort data=_anal14 out=_catby14(keep=_byvar1) nodupkey;
  by _byvar1;
  where paramcd eq upcase("Maxsmp") and ex_none_flg=0 and eventfl="Y";
run;

data _temp14;
  set _anal14;
  output;
run;

proc sort data=_temp14 out=_temp914 nodupkey;
  by _datasrt _byvar1 _blcksrt _cat AVAL _trt usubjid;
  where paramcd eq upcase("Maxsmp") and ex_none_flg=0 and eventfl="Y";
run;

proc freq data=_temp914 noprint;
  format AVAL;
  tables _datasrt*_byvar1*_blcksrt*_cat * AVAL * _trt / sparse norow nocol
    nopercnt out=_pct14(drop=percent);
run;

proc sort data=_temp14 out=_analcnt14 nodupkey;
  by _datasrt _byvar1 _cat _trt USUBJID;
  where paramcd eq upcase("Maxsmp") and knowvfl eq 'Y';
run;

proc freq data=_analcnt14 noprint;
  tables _datasrt*_byvar1*_cat * _trt / sparse noprint
    out=_denom14(drop=percent);
run;

data _denomf14;
  length _cat $100;
  _datasrt=1;
  set _catby14(keep=_byvar1);
  * All treatment groups ;
  _trt1=0;
  _trt2=0;
  * _CAT is the subgroup variable ;
  _cat="12-15 Years ";
  output;
  _cat="16-25 Years ";
  output;
run;

proc transpose data=_denom14 out=_denomin14(drop=_name _label_) prefix=_trt;
  by _datasrt _byvar1 _cat;
  var count;
  id _trt;
run;

```

```

data _frame14;
    set _frame4;
    _blcksrt=7;
run;

proc sort data=_frame14;
    by _datasrt _byvar1 _blcksrt _cat AVAL _trt;
run;

proc sort data=_pct14;
    by _datasrt _byvar1 _blcksrt _cat AVAL _trt;
run;

data _pct14;
    merge _frame14(in=_inframe) _pct14;
    by _datasrt _byvar1 _blcksrt _cat AVAL _trt;

    if _inframe;

    if count=. then
        count=0;
run;

proc sort data=_pct14;
    by _datasrt _byvar1 _blcksrt AVAL;
run;

data _miss14(keep=_datasrt _byvar1 _blcksrt AVAL totcount);
    set _pct14;
    where AVAL=9998;
    retain totcount;
    by _datasrt _byvar1 _blcksrt AVAL;

    if first.AVAL then
        totcount=0;
    totcount=totcount+count;

    if last.AVAL;
run;

data _pct14(drop=totcount);
    merge _pct14 _miss14;
    by _datasrt _byvar1 _blcksrt AVAL;

    if totcount=0 then
        delete;
run;

proc sort data=_denomf14;
    by _datasrt _byvar1 _cat;
run;

proc sort data=_denomin14;

```

```

    by _datasrt _byvar1 _cat;
run;

data _denomin14;
    merge _denomf14(in=_inframe) _denomin14;
    by _datasrt _byvar1 _cat;

    if _inframe;
    _blcksrt=7;
run;

proc sort data=_pct14;
    by _datasrt _byvar1 _cat;
run;

data _pct14;
    if 0 then
        set _basetemplate;
    merge _denomin14(in=_a) _pct14;
    by _datasrt _byvar1 _cat;

    if _a;
    _varname="AVAL ";
    _vrlabel=" ";
    _rwlabel=put(AVAL, sev.);

    if AVAL=9998 then
        do;
            _rwlabel="Missing ";
            _catord=9998;
        end;
    else if AVAL=9999 then
        do;
            _rwlabel="Total ";
            _catord=9999;
        end;

    if _catord=. then
        _catord=9997;
run;

proc sort data=_pct14;
    by _datasrt _byvar1 _blcksrt _catord AVAL _trt _cat;
run;

data _base14;
    length _catlabl $200;
    set _pct14 end=eof;
    by _datasrt _byvar1 _blcksrt _catord AVAL _trt _cat;
    retain _rowsrt 6 _rowmax 0;
    array _trtcnt(*) _trt1-_trt3;
    drop _rowmax _cpct;
    length _cpct $100;
    _cpct='';

```



```

_module='mcatstat';

if count > . then
    _cvalue=put(count, 5.);
else
    _cvalue=put(0, 5.);
*-----;
* Format percent to append to display value in _CVALUE ;
*-----;

if _trt ne . then
    do;

        if _trtcnt(_trt) > 0 then
            do;
                percent=count / _trtcnt(_trt) * 100;

                if percent > 0 then
                    do;

                        if round(percent, 0.1) GE 0.1 then
                            _cpct="(*ESC*){nbspspace 1}("||strip(put(percent, 5.1))||")";
                        else
                            _cpct="(*ESC*){nbspspace 1}(0.0)";
                        _cvalue=trim(_cvalue)||_cpct;
                    end;
                end;
            end;
        end;

    if length(_cvalue) < 13 then
        do;
            *-----;
            * Put character A0x at right most character to pad text;
            *-----;
            substr(_cvalue, 13, 1)='A0'x;
        end;

    if first._byvar1 then
        _rowsrt=6;

    if first.AVAL then
        do;
            _rowsrt=_rowsrt + 1;
            _rowmax=max(_rowsrt, _rowmax);
        end;
    _datatype='data';
    _indent=0;
    _dptindt=0;
    _vorder=1;
    _rowjump=1;

    if upcase(_rwlabel)='_ NONE _' then
        _rwlabel=' ';
    _indent=8;

```

```

_dptindt=0;

if _trt=2+1 then
    _trt=9999;

if eof then
    call symput('_rowsrt', compress(put(_rowmax, 4.)));
_direct="TOP ";
_p=2;
run;

data _base14;
    set _base14;
    length _cvalue2 $30.;
    _cvalue2=strip(tranwrd(_cvalue, 'A0'x, ""));
    _cvalue21=strip(scan(_cvalue, 1, '('));
    _cvalue22=compress(scan(_cvalue, 2, '(', ')'));
run;

data _base13;
    set _base13;
    drop _trt1 _trt2 count;
run;

proc sort data=_base13;
    by _datasrt _byvar1 _cat _trt;
run;

data _base13;
    set _base13;

    if _trt=1 then
        do;
            _trt1=input(_cvalue, ??best.);
        end;

    if _trt=2 then
        do;
            _trt2=input(_cvalue, ??best.);
        end;
run;

proc sort data=_base14(keep=_datasrt _trt _cvalue2 _cvalue21 _cvalue22 _cat
    _byvar1 count);
    by _datasrt _byvar1 _cat _trt;
quit;

data _base14;
    merge _base13(in=a) _base14(in=b);
    by _datasrt _byvar1 _cat _trt;

    if a;

    if a and not b then

```

```

do;
    _cvalue2="0";
    _cvalue21="0";
end;

if compress(_cvalue2)="0" then
    _cvalue22=put(0, 5.1);

if compress(_cvalue)="0" then
do;
    _cvalue2="NA";
    _cvalue21="NA";
    _cvalue22="NA";
end;

run;

data _base13;
    set _base13;
    delete;
run;

*****
* Specification 5.7.3: Calculate 95% CI for observed proportion          *;
*****

data _cnp _tmp_cnp;
    set _base14;

    if count=. then
        count=0;
    indc=1;
    output _cnp;
    indc=2;

    if _trt=1 then
        do;
            count=_trt1 - count;
        end;

    if _trt=2 then
        do;
            count=_trt2 - count;
        end;
    output _cnp;

    if indc=2 and count=0 then
        output _tmp_cnp;

run;

proc sort data=_cnp;
    by _byvar1 _cat _rowsrt _rwlabel _trt;
run;

proc sort nodupkey data=_tmp_cnp(keep=_byvar1 _cat _rowsrt _rwlabel _trt);

```

```

by _byvar1 _cat _rowsrt _rwlabel _trt;
run;

*****
* Call proc freq procedure to calculate CI for observed proportion      *;
*****
proc freq data=_cnp noprint;
  by _byvar1 _cat _rowsrt _rwlabel _trt;
  table indc/binomial alpha=0.05;
  output out=obsprop binomial;
  weight count;
run;

data obsprop;
  merge obsprop _tmp_cnp(in=a);
  by _byvar1 _cat _rowsrt _rwlabel _trt;

  if _bin_=1 and not a then
    do;
      xl_bin_=1 - xu_bin;
      xu_bin_=1 - xl_bin;
    end;
  else
    do;
      xl_bin_=xl_bin;
      xu_bin_=xu_bin;
    end;
run;

data cnpobsprop1(keep=_byvar1 _cat _rowsrt _rwlabel _trt cnp_ci);
  set obsprop;
  by _byvar1 _cat _rowsrt _rwlabel _trt;
  cnp_ci=(' || compress(put(xl_bin_ * 100, 5.1))
    || '(*ESC*){nbsp 1}' || compress(put(xu_bin_ * 100, 5.1)) || ');
  label cnp_ci='95% CI';
run;

proc datasets lib=work nolist gennum=all;
  delete _cnp obsprop;
run;

proc sort data=_base14;
  by _byvar1 _cat _rowsrt _rwlabel _trt;
run;

proc sort data=cnpobsprop1;
  by _byvar1 _cat _rowsrt _rwlabel _trt;
run;

data _base14;
  merge _base14(in=a) cnpobsprop1;
  by _byvar1 _cat _rowsrt _rwlabel _trt;

  if a;

```

```

        if compress(_cvalue)="0" then
            do;
                cnp_ci="NE";
            end;
run;

*****
* Specification 5.8: Statistics for New or worsened joint pain category      *;
*****
* Specification 5.8.1: Count denominator (N)                               *;
*****

data _anal15;
    set _anal1;
    _blcksrt=8;
run;

proc sort data=_anal15;
    by _datasrt _byvar1 _blcksrt DENOMFL _trt _cat;
run;

proc sort data=_anal15 out=_catby15(keep=_byvar1) nodupkey;
    by _byvar1;
    where paramcd eq upcase("Maxsjp");
run;

data _temp15;
    set _anal15;
    output;
run;

proc sort data=_temp15 out=_temp915 nodupkey;
    by _datasrt _byvar1 _blcksrt _cat DENOMFL _trt usubjid;
    where paramcd eq upcase("Maxsjp");
run;

proc freq data=_temp915 noprint;
    format DENOMFL;
    tables _datasrt*_byvar1*_blcksrt*_cat * DENOMFL * _trt / sparse norow nocol
        nopercnt out=_pct15(drop=percent);
run;

proc freq data=_pct15 noprint;
    where DENOMFL ne 9999;
    weight count;
    tables _datasrt*_byvar1*_cat * _trt / sparse noprint
        out=_denom15(drop=percent);
run;

data _denomf15;
    length _cat $100;
    _datasrt=1;
    set _catby15(keep=_byvar1);

```

```

* All treatment groups ;
_trt1=0;
_trt2=0;
* _CAT is the subgroup variable ;
_cat="12-15 Years ";
output;
_cat="16-25 Years ";
output;
run;

proc transpose data=_denom15 out=_denomin15(drop=_name __label_) prefix=_trt;
  by _datasrt _byvar1 _cat;
  var count;
  id _trt;
run;

data _frame15;
  set _frame3;
  _blcksrt=8;
run;

proc sort data=_frame15;
  by _datasrt _byvar1 _blcksrt _cat DENOMFL _trt;
run;

proc sort data=_pct15;
  by _datasrt _byvar1 _blcksrt _cat DENOMFL _trt;
run;

data _pct15;
  merge _frame15(in=_inframe) _pct15;
  by _datasrt _byvar1 _blcksrt _cat DENOMFL _trt;

  if _inframe;

  if count=. then
    count=0;
run;

proc sort data=_pct15;
  by _datasrt _byvar1 _blcksrt DENOMFL;
run;

data _miss15(keep=_datasrt _byvar1 _blcksrt DENOMFL totcount);
  set _pct15;
  where DENOMFL=9998;
  retain totcount;
  by _datasrt _byvar1 _blcksrt DENOMFL;

  if first.DENOMFL then
    totcount=0;
  totcount=totcount+count;

```

```

        if last.DENOMFL;
run;

data _pct15(drop=totcount);
    merge _pct15 _miss15;
    by _datasrt _byvar1 _blcksrt DENOMFL;

    if totcount=0 then
        delete;
run;

proc sort data=_denomf15;
    by _datasrt _byvar1 _cat;
run;

proc sort data=_denomin15;
    by _datasrt _byvar1 _cat;
run;

data _denomin15;
    merge _denomf15(in=_inframe) _denomin15;
    by _datasrt _byvar1 _cat;

    if _inframe;
        _blcksrt=8;
run;

proc sort data=_pct15;
    by _datasrt _byvar1 _cat;
run;

data _pct15;
    if 0 then
        set _basetemplate;
    merge _denomin15(in=_a) _pct15;
    by _datasrt _byvar1 _cat;

    if _a;
        _varname="DENOMFL ";
        _vrlabel="New or worsened joint pain(*ESC*){super d} ";
        _rwlabel=put(DENOMFL, sev.);

    if DENOMFL=9998 then
        do;
            _rwlabel="Missing ";
            _catord=9998;
        end;
    else if DENOMFL=9999 then
        do;
            _rwlabel="Total ";
            _catord=9999;
        end;

    if _catord=. then

```

```

        _catord=9997;
run;

proc sort data=_pct15;
    by _datasrt _byvar1 _blcksrt _catord DENOMFL _trt _cat;
run;

data _base15;
    length _catlabl $200;
    set _pct15 end=eof;
    by _datasrt _byvar1 _blcksrt _catord DENOMFL _trt _cat;
    retain _rowsrt 0 _rowmax 0;
    array _trtcnt(*) _trt1-_trt3;
    drop _rowmax _cpct;
    length _cpct $100;
    _cpct=' ';
    _module='mcatstat';

    if count > . then
        _cvalue=put(count, 5.);
    else
        _cvalue=put(0, 5.);

    if length(_cvalue) < 5 then
        do;
            *-----;
            * Put character A0x at right most character to pad text;
            *-----;
            substr(_cvalue, 5, 1)='A0'x;
        end;

    if first._byvar1 then
        _rowsrt=0;

    if first.DENOMFL then
        do;
            _rowsrt=_rowsrt + 1;
            _rowmax=max(_rowsrt, _rowmax);
        end;
    _datatyp='data';
    _indent=0;
    _dptindt=0;
    _vorder=1;
    _rowjump=1;

    if upcase(_rwlabel)='_NONE_' then
        _rwlabel=' ';
    _indent=8;
    _dptindt=0;

    if _trt=2 +1 then
        _trt=9999;

    if eof then

```



```

        call symput('_rowsrt', compress(put(_rowmax, 4.)));
        _direct="TOP ";
        _p=0;
run;

*****
* Specification 5.8.2: Count n and percentage (%) for individual severity      *;
*****

data _anal16;
    set _anal2;
    _blcksrt=8;
run;

proc sort data=_anal16;
    by _datasrt _byvar1 _blcksrt AVAL _trt _cat;
run;

proc sort data=_anal16 out=_catby16(keep=_byvar1) nodupkey;
    by _byvar1;
    where paramcd eq upcase("Maxsjp") and ex_none_flg=0 and eventfl="Y";
run;

data _temp16;
    set _anal16;
    output;
run;

proc sort data=_temp16 out=_temp916 nodupkey;
    by _datasrt _byvar1 _blcksrt _cat AVAL _trt usubjid;
    where paramcd eq upcase("Maxsjp") and ex_none_flg=0 and eventfl="Y";
run;

proc freq data=_temp916 noprint;
    format AVAL;
    tables _datasrt*_byvar1*_blcksrt*_cat * AVAL * _trt / sparse norow nocol
           nopercnt out=_pct16(drop=percent);
run;

proc sort data=_temp16 out=_analcnt16 nodupkey;
    by _datasrt _byvar1 _cat _trt USUBJID;
    where paramcd eq upcase("Maxsjp") and knowvfl eq 'Y';
run;

proc freq data=_analcnt16 noprint;
    tables _datasrt*_byvar1*_cat * _trt / sparse noprint
           out=_denom16(drop=percent);
run;

data _denomf16;
    length _cat $100;
    _datasrt=1;
    set _catby16(keep=_byvar1);
    * All treatment groups ;

```

```

    _trt1=0;
    _trt2=0;
    * _CAT is the subgroup variable ;
    _cat="12-15 Years ";
    output;
    _cat="16-25 Years ";
    output;
run;

proc transpose data=_denom16 out=_denomin16(drop=_name_ _label_) prefix=_trt;
  by _datasrt _byvar1 _cat;
  var count;
  id _trt;
run;

data _frame16;
  set _frame4;
  _blcksrt=8;
run;

proc sort data=_frame16;
  by _datasrt _byvar1 _blcksrt _cat AVAL _trt;
run;

proc sort data=_pct16;
  by _datasrt _byvar1 _blcksrt _cat AVAL _trt;
run;

data _pct16;
  merge _frame16(in=_inframe) _pct16;
  by _datasrt _byvar1 _blcksrt _cat AVAL _trt;

  if _inframe;

  if count=. then
    count=0;
run;

proc sort data=_pct16;
  by _datasrt _byvar1 _blcksrt AVAL;
run;

data _miss16(keep=_datasrt _byvar1 _blcksrt AVAL totcount);
  set _pct16;
  where AVAL=9998;
  retain totcount;
  by _datasrt _byvar1 _blcksrt AVAL;

  if first.AVAL then
    totcount=0;
  totcount=totcount+count;

  if last.AVAL;
run;

```

```

data _pct16(drop=totcount);
  merge _pct16 _miss16;
  by _datasrt _byvar1 _blcksrt AVAL;

  if totcount=0 then
    delete;
run;

proc sort data=_denomf16;
  by _datasrt _byvar1 _cat;
run;

proc sort data=_denomin16;
  by _datasrt _byvar1 _cat;
run;

data _denomin16;
  merge _denomf16(in=_inframe) _denomin16;
  by _datasrt _byvar1 _cat;

  if _inframe;
  _blcksrt=8;
run;

proc sort data=_pct16;
  by _datasrt _byvar1 _cat;
run;

data _pct16;
  if 0 then
    set _basemplate;
  merge _denomin16(in=_a) _pct16;
  by _datasrt _byvar1 _cat;

  if _a;
  _varname="AVAL ";
  _vrlabel=" ";
  _rwlabel=put(AVAL, sev.);

  if AVAL=9998 then
    do;
      _rwlabel="Missing ";
      _catord=9998;
    end;
  else if AVAL=9999 then
    do;
      _rwlabel="Total ";
      _catord=9999;
    end;

  if _catord=. then
    _catord=9997;
run;

```

```

proc sort data=_pct16;
  by _datasrt _byvar1 _blcksrt _catord AVAL _trt _cat;
run;

data _base16;
  length _catlabl $200;
  set _pct16 end=eof;
  by _datasrt _byvar1 _blcksrt _catord AVAL _trt _cat;
  retain _rowsrt 6 _rowmax 0;
  array _trtcnt(*) _trt1-_trt3;
  drop _rowmax _cpct;
  length _cpct $100;
  _cpct='';
  _module='mcatstat';

  if count > . then
    _cvalue=put(count, 5.);
  else
    _cvalue=put(0, 5.);
  *-----;
  * Format percent to append to display value in _CVALUE ;
  *-----;

  if _trt ne . then
    do;

      if _trtcnt(_trt) > 0 then
        do;
          percent=count / _trtcnt(_trt) * 100;

          if percent > 0 then
            do;

              if round(percent, 0.1) GE 0.1 then
                _cpct="(*ESC*){nbspspace 1}("||strip(put(percent, 5.1))||")";
              else
                _cpct="(*ESC*){nbspspace 1}{(0.0)";
              _cvalue=trim(_cvalue)||_cpct;
            end;
          end;
        do;

      if length(_cvalue) < 13 then
        do;
          *-----;
          * Put character A0x at right most character to pad text;
          *-----;
          substr(_cvalue, 13, 1)='A0'x;
        end;

      if first._byvar1 then
        _rowsrt=6;

```

```

if first.AVAL then
    do;
        _rowsrt=_rowsrt + 1;
        _rowmax=max(_rowsrt, _rowmax);
    end;
    _datatype='data';
    _indent=0;
    _dptindt=0;
    _vorder=1;
    _rowjump=1;

if upcase(_rwlabel)='_NONE_' then
    _rwlabel=' ';
    _indent=8;
    _dptindt=0;

if _trt=2 +1 then
    _trt=9999;

if eof then
    call symput('_rowsrt', compress(put(_rowmax, 4.)));
    _direct="TOP ";
    _p=2;
run;

data _base16;
    set _base16;
    length _cvalue2 $30.;
    _cvalue2=strip(tranwrd(_cvalue, 'A0'x, ""));
    _cvalue21=strip(scan(_cvalue, 1, '('));
    _cvalue22=compress(scan(_cvalue, 2, '(', ')'));
run;

data _base15;
    set _base15;
    drop _trt1 _trt2 count;
run;

proc sort data=_base15;
    by _datasrt _byvar1 _cat _trt;
run;

data _base15;
    set _base15;

    if _trt=1 then
        do;
            _trt1=input(_cvalue, ??best.);
        end;

    if _trt=2 then
        do;
            _trt2=input(_cvalue, ??best.);
        end;

```

```
run;

proc sort data=_base16(keep=_datasrt _trt _cvalue2 _cvalue21 _cvalue22 _cat
    _byvar1 count);
    by _datasrt _byvar1 _cat _trt;
quit;
```

```
data _base16;
    merge _base15(in=a) _base16(in=b);
    by _datasrt _byvar1 _cat _trt;

    if a;

    if a and not b then
        do;
            _cvalue2="0";
            _cvalue21="0";
        end;

    if compress(_cvalue2)="0" then
        _cvalue22=put(0, 5.1);

    if compress(_cvalue)="0" then
        do;
            _cvalue2="NA";
            _cvalue21="NA";
            _cvalue22="NA";
        end;
```

```
run;
```

```
data _base15;
    set _base15;
    delete;
```

```
run;
```

```
*****;
* Specification 5.8.3: Calculate 95% CI for observed proportion *;
*****;
```

```
data _cnp _tmp_cnp;
    set _base16;

    if count=. then
        count=0;
    indc=1;
    output _cnp;
    indc=2;

    if _trt=1 then
        do;
            count=_trt1 - count;
        end;

    if _trt=2 then
```

```

do;
    count=_trt2 - count;
end;
output _cnp;

if indc=2 and count=0 then
    output _tmp_cnp;
run;

proc sort data=_cnp;
    by _byvar1 _cat _rowsrt _rwlabel _trt;
run;

proc sort nodupkey data=_tmp_cnp(keep=_byvar1 _cat _rowsrt _rwlabel _trt);
    by _byvar1 _cat _rowsrt _rwlabel _trt;
run;

*****
* Call proc freq procedure to calculate CI for observed proportion          *;
*****;

proc freq data=_cnp noprint;
    by _byvar1 _cat _rowsrt _rwlabel _trt;
    table indc/binomial alpha=0.05;
    output out=obsprop binomial;
    weight count;
run;

data obsprop;
    merge obsprop _tmp_cnp(in=a);
    by _byvar1 _cat _rowsrt _rwlabel _trt;

    if _bin_=1 and not a then
        do;
            xl_bin_=1 - xu_bin;
            xu_bin_=1 - xl_bin;
        end;
    else
        do;
            xl_bin_=xl_bin;
            xu_bin_=xu_bin;
        end;
run;

data cnpobsprop1(keep=_byvar1 _cat _rowsrt _rwlabel _trt cnp_ci);
    set obsprop;
    by _byvar1 _cat _rowsrt _rwlabel _trt;
    cnp_ci=(' || compress(put(xl_bin_ * 100, 5.1))
            || '(*ESC*){nbspace 1}' || compress(put(xu_bin_ * 100, 5.1)) || ');
    label cnp_ci='95% CI';
run;

proc datasets lib=work nolist gennum=all;
    delete _cnp obsprop;
run;

```

```

proc sort data=_base16;
  by _byvar1 _cat _rowsrt _rwlabel _trt;
run;

proc sort data=cnpobsprop1;
  by _byvar1 _cat _rowsrt _rwlabel _trt;
run;

data _base16;
  merge _base16(in=a) cnpobsprop1;
  by _byvar1 _cat _rowsrt _rwlabel _trt;

  if a;

  if compress(_cvalue)="0" then
    do;
      cnp_ci="NE";
    end;
run;

*****
* Specification 5.9: Statistics for Any systemic event category      *;
*****
* Specification 5.9.1: Count denominator (N)                        *;
*****

data _anal17;
  set _anal1;
  _blcksrt=9;
run;

proc sort data=_anal17;
  by _datasrt _byvar1 _blcksrt DENOMFL _trt _cat;
run;

proc sort data=_anal17 out=_catby17(keep=_byvar1) nodupkey;
  by _byvar1;
  where paramcd eq upcase("Any");
run;

data _temp17;
  set _anal17;
  output;
run;

proc sort data=_temp17 out=_temp917 nodupkey;
  by _datasrt _byvar1 _blcksrt _cat DENOMFL _trt usubjid;
  where paramcd eq upcase("Any");
run;

proc freq data=_temp917 noprint;
  format DENOMFL;
  tables _datasrt*_byvar1*_blcksrt*_cat * DENOMFL * _trt / sparse norow nocol

```

FDA-CBER-2022-5812-0735799



```

        nopercnt out=_pct17(drop=percent);
run;

proc freq data=_pct17 noprint;
  where DENOMFL ne 9999;
  weight count;
  tables _datasrt*_byvar1*_cat * _trt / sparse noprint
        out=_denom17(drop=percent);
run;

data _denomf17;
  length _cat $100;
  _datasrt=1;
  set _catby17(keep=_byvar1);
  * All treatment groups ;
  _trt1=0;
  _trt2=0;
  * _CAT is the subgroup variable ;
  _cat="12-15 Years ";
  output;
  _cat="16-25 Years ";
  output;
run;

proc transpose data=_denom17 out=_denomin17(drop=_name__label_) prefix=_trt;
  by _datasrt _byvar1 _cat;
  var count;
  id _trt;
run;

data _frame17;
  _datasrt=1;
  set _catby17(keep=_byvar1);
  _blcksrt=9;
  length DENOMFL 8;
  length _cat $100;
  _catLabl=" ";
  _trt=1;
  DENOMFL=1;
  _catord=1;
  _subcat=1;
  _cat="12-15 Years ";
  output;
  _subcat=2;
  _cat="16-25 Years ";
  output;
  _trt=2;
  DENOMFL=1;
  _catord=1;
  _subcat=1;
  _cat="12-15 Years ";
  output;
  _subcat=2;
  _cat="16-25 Years ";

```

```

        output;
run;

proc sort data=_frame17;
    by _datasrt _byvar1 _blcksrt _cat DENOMFL _trt;
run;

proc sort data=_pct17;
    by _datasrt _byvar1 _blcksrt _cat DENOMFL _trt;
run;

data _pct17;
    merge _frame17(in=_inframe) _pct17;
    by _datasrt _byvar1 _blcksrt _cat DENOMFL _trt;

    if _inframe;

    if count=. then
        count=0;
run;

proc sort data=_pct17;
    by _datasrt _byvar1 _blcksrt DENOMFL;
run;

data _miss17(keep=_datasrt _byvar1 _blcksrt DENOMFL totcount);
    set _pct17;
    where DENOMFL=9998;
    retain totcount;
    by _datasrt _byvar1 _blcksrt DENOMFL;

    if first.DENOMFL then
        totcount=0;
    totcount=totcount+count;

    if last.DENOMFL;
run;

data _pct17(drop=totcount);
    merge _pct17 _miss17;
    by _datasrt _byvar1 _blcksrt DENOMFL;

    if totcount=0 then
        delete;
run;

proc sort data=_denomf17;
    by _datasrt _byvar1 _cat;
run;

proc sort data=_denomin17;
    by _datasrt _byvar1 _cat;
run;

```

```

data _denomin17;
  merge _denomfl17(in=_inframe) _denomin17;
  by _datasrt _byvar1 _cat;

  if _inframe;
  _blcksrt=9;
run;

proc sort data=_pct17;
  by _datasrt _byvar1 _cat;
run;

data _pct17;
  if 0 then
    set _basetemplate;
  merge _denomin17(in=_a) _pct17;
  by _datasrt _byvar1 _cat;

  if _a;
  _varname="DENOMFL ";
  _vrlabel=" ";
  _rwlabel="Any systemic event(*ESC*){super g} ";

  if DENOMFL=9998 then
    do;
      _rwlabel="Missing ";
      _catord=9998;
    end;
  else if DENOMFL=9999 then
    do;
      _rwlabel="Total ";
      _catord=9999;
    end;

  if _catord=. then
    _catord=9997;
run;

proc sort data=_pct17;
  by _datasrt _byvar1 _blcksrt _catord DENOMFL _trt _cat;
run;

data _base17;
  length _catlabl $200;
  set _pct17 end=eof;
  by _datasrt _byvar1 _blcksrt _catord DENOMFL _trt _cat;
  retain _rowsrt 0 _rowmax 0;
  array _trtcnt(*) _trt1-_trt3;
  drop _rowmax _cpct;
  length _cpct $100;
  _cpct='';
  _module='mcatstat';

  if count > . then

```

```

        _cvalue=put(count, 5.);
else
    _cvalue=put(0, 5.);

if length(_cvalue) < 5 then
    do;
        *-----;
        * Put character A0x at right most character to pad text;
        *-----;
        substr(_cvalue, 5, 1)='A0'x;
    end;

if first._byvar1 then
    _rowsrt=0;

if first.DENOMFL then
    do;
        _rowsrt=_rowsrt + 1;
        _rowmax=max(_rowsrt, _rowmax);
    end;
    _datatype='data';
    _indent=0;
    _dptindt=0;
    _vorder=1;
    _rowjump=1;

if upcase(_rwlabel)='_NONE_' then
    _rwlabel=' ';
    _indent=0;
    _dptindt=0;

if _trt=2 +1 then
    _trt=9999;

if eof then
    call symput('_rowsrt', compress(put(_rowmax, 4.)));
    _direct="TOP ";
    _p=0;
run;

*****
* Specification 5.9.2: Count n and percentage (%) for individual severity    *;
*****

data _anal18;
    set _anal2;
    _blcksrt=9;
    _cnt=1;
run;

proc sort data=_anal18;
    by _datasrt _byvar1 _blcksrt AVAL _trt _cat;
run;

```

```

proc sort data=_anal18 out=_catby18(keep=_byvar1) nodupkey;
  by _byvar1;
  where paramcd eq upcase("Any") and ex_none_flg=0 and _any_flg=1 and
    eventfl="Y";
run;

data _temp18;
  set _anal18;
  output;
run;

proc sort data=_temp18 out=_temp918 nodupkey;
  by _datasrt _byvar1 _blcksrt _cat AVAL _trt usubjid;
  where paramcd eq upcase("Any") and ex_none_flg=0 and _any_flg=1 and
    eventfl="Y";
run;

proc freq data=_temp918 noprint;
  format AVAL;
  tables _datasrt*_byvar1*_blcksrt*_cat * AVAL * _trt / sparse norow nocol
    nopercnt out=_pct18(drop=percent);
run;

proc sort data=_temp18 out=_analcnt18 nodupkey;
  by _datasrt _byvar1 _cat _trt USUBJID;
  where paramcd eq upcase("Any") and knowvfl eq 'Y';
run;

proc freq data=_analcnt18 noprint;
  tables _datasrt*_byvar1*_cat * _trt / sparse noprint
    out=_denom18(drop=percent);
run;

data _denomf18;
  length _cat $100;
  _datasrt=1;
  set _catby18(keep=_byvar1);
  * All treatment groups ;
  _trt1=0;
  _trt2=0;
  * _CAT is the subgroup variable ;
  _cat="12-15 Years ";
  output;
  _cat="16-25 Years ";
  output;
run;

proc transpose data=_denom18 out=_denomin18(drop=_name _label_) prefix=_trt;
  by _datasrt _byvar1 _cat;
  var count;
  id _trt;
run;

```

```

data _frame18;
  _datasrt=1;
  set _catby18(keep=_byvar1);
  _blcksrt=9;
  length AVAL 8;
  length _cat $100;
  _catLabl=" ";
  _trt=1;
  AVAL=1;
  _catord=1;
  _subcat=1;
  _cat="12-15 Years ";
  output;
  _subcat=2;
  _cat="16-25 Years ";
  output;
  _trt=2;
  AVAL=1;
  _catord=1;
  _subcat=1;
  _cat="12-15 Years ";
  output;
  _subcat=2;
  _cat="16-25 Years ";
  output;
run;

proc sort data=_frame18;
  by _datasrt _byvar1 _blcksrt _cat AVAL _trt;
run;

proc sort data=_pct18;
  by _datasrt _byvar1 _blcksrt _cat AVAL _trt;
run;

data _pct18;
  merge _frame18(in=_inframe) _pct18;
  by _datasrt _byvar1 _blcksrt _cat AVAL _trt;

  if _inframe;

  if count=. then
    count=0;
run;

proc sort data=_pct18;
  by _datasrt _byvar1 _blcksrt AVAL;
run;

data _miss18(keep=_datasrt _byvar1 _blcksrt AVAL totcount);
  set _pct18;
  where AVAL=9998;
  retain totcount;

```

```

by _datasrt _byvar1 _blcksrt AVAL;

if first.AVAL then
    totcount=0;
totcount=totcount+count;

if last.AVAL;
run;

data _pct18(drop=totcount);
merge _pct18 _miss18;
by _datasrt _byvar1 _blcksrt AVAL;

if totcount=0 then
    delete;
run;

proc sort data=_denomf18;
by _datasrt _byvar1 _cat;
run;

proc sort data=_denomin18;
by _datasrt _byvar1 _cat;
run;

data _denomin18;
merge _denomf18(in=_inframe) _denomin18;
by _datasrt _byvar1 _cat;

if _inframe;
    _blcksrt=9;
run;

proc sort data=_pct18;
by _datasrt _byvar1 _cat;
run;

data _pct18;
if 0 then
    set _basemplate;
merge _denomin18(in=_a) _pct18;
by _datasrt _byvar1 _cat;

if _a;
    _varname="AVAL ";
    _vrlabel=" ";
    _rwlabel="Any systemic event(*ESC*){super g} ";

if AVAL=9998 then
    do;
        _rwlabel="Missing ";
        _catord=9998;
    end;
else if AVAL=9999 then

```

```

do;
    _rwlablel="Total ";
    _catord=9999;
end;

if _catord=. then
    _catord=9997;

run;

proc sort data=_pct18;
    by _datasrt _byvar1 _blcksrt _catord AVAL _trt _cat;
run;

data _base18;
    length _catlabl $200;
    set _pct18 end=eof;
    by _datasrt _byvar1 _blcksrt _catord AVAL _trt _cat;
    retain _rowsrt 1 _rowmax 0;
    array _trtcnt(*) _trt1-_trt3;
    drop _rowmax _cpct;
    length _cpct $100;
    _cpct='';
    _module='mcatstat';

    if count > . then
        _cvalue=put(count, 5.);
    else
        _cvalue=put(0, 5.);
    *-----;
    * Format percent to append to display value in _CVALUE ;
    *-----;

    if _trt ne . then
        do;

            if _trtcnt(_trt) > 0 then
                do;
                    percent=count / _trtcnt(_trt) * 100;

                    if percent > 0 then
                        do;

                            if round(percent, 0.1) GE 0.1 then
                                _cpct="(*ESC*){nbspspace 1}("||strip(put(percent, 5.1))||")";
                            else
                                _cpct="(*ESC*){nbspspace 1}(0.0)";
                            _cvalue=trim(_cvalue)||_cpct;
                        end;
                    end;
                end;

            end;

        end;

    if length(_cvalue) < 13 then
        do;
            *-----;

```



```

        * Put character A0x at right most character to pad text;
        *-----;
        substr(_cvalue, 13, 1)='A0'x;
    end;

    if first._byvar1 then
        _rowsrt=1;

    if first.AVAL then
        do;
            _rowsrt=_rowsrt + 1;
            _rowmax=max(_rowsrt, _rowmax);
        end;
        _datatype='data';
        _indent=0;
        _dptindt=0;
        _vorder=1;
        _rowjump=1;

    if upcase(_rwlabel)='_NONE_' then
        _rwlabel=' ';
        _indent=0;
        _dptindt=0;

    if _trt=2 +1 then
        _trt=9999;

    if eof then
        call symput('_rowsrt', compress(put(_rowmax, 4.)));
        _direct="TOP ";
        _p=2;
run;

data _base18;
    set _base18;
    length _cvalue2 $30.;
    _cvalue2=strip(tranwrd(_cvalue, 'A0'x, ""));
    _cvalue21=strip(scan(_cvalue, 1, '('));
    _cvalue22=compress(scan(_cvalue, 2, '(', ')'));
run;

data _base17;
    set _base17;
    drop _trt1 _trt2 count;
run;

proc sort data=_base17;
    by _datasrt _byvar1 _cat _trt;
run;

data _base17;
    set _base17;

    if _trt=1 then

```

```

do;
  _trt1=input(_cvalue, ??best.);
end;

if _trt=2 then
  do;
    _trt2=input(_cvalue, ??best.);
  end;
run;

proc sort data=_base18(keep=_datasrt _trt _cvalue2 _cvalue21 _cvalue22 _cat
  _byvar1 count);
  by _datasrt _byvar1 _cat _trt;
quit;

```

```

data _base18;
  merge _base17(in=a) _base18(in=b);
  by _datasrt _byvar1 _cat _trt;

  if a;

  if a and not b then
    do;
      _cvalue2="0";
      _cvalue21="0";
    end;

  if compress(_cvalue2)="0" then
    _cvalue22=put(0, 5.1);

  if compress(_cvalue)="0" then
    do;
      _cvalue2="NA";
      _cvalue21="NA";
      _cvalue22="NA";
    end;
run;

```

```

data _base17;
  set _base17;
  delete;
run;

```

```

*****
* Specification 5.9.3: Calculate 95% CI for observed proportion      *;
*****

```

```

data _cnp _tmp_cnp;
  set _base18;

  if count=. then
    count=0;
  indc=1;
  output _cnp;

```

```

indc=2;

if _trt=1 then
  do;
    count=_trt1 - count;
  end;

if _trt=2 then
  do;
    count=_trt2 - count;
  end;
output _cnp;

if indc=2 and count=0 then
  output _tmp_cnp;
run;

proc sort data=_cnp;
  by _byvar1 _cat _rowsrt _rwlabel _trt;
run;

proc sort nodupkey data=_tmp_cnp(keep=_byvar1 _cat _rowsrt _rwlabel _trt);
  by _byvar1 _cat _rowsrt _rwlabel _trt;
run;

*****
* Call proc freq procedure to calculate CI for observed proportion      *;
*****
proc freq data=_cnp noprint;
  by _byvar1 _cat _rowsrt _rwlabel _trt;
  table indc/binomial alpha=0.05;
  output out=obsprop binomial;
  weight count;
run;

data obsprop;
  merge obsprop _tmp_cnp(in=a);
  by _byvar1 _cat _rowsrt _rwlabel _trt;

  if _bin_=1 and not a then
    do;
      xl_bin_=1 - xu_bin;
      xu_bin_=1 - xl_bin;
    end;
  else
    do;
      xl_bin_=xl_bin;
      xu_bin_=xu_bin;
    end;
run;

data cnpobsprop1(keep=_byvar1 _cat _rowsrt _rwlabel _trt cnp_ci);
  set obsprop;
  by _byvar1 _cat _rowsrt _rwlabel _trt;

```

```

cnp_ci=(' || compress(put(xl_bin_ * 100, 5.1))
|| '(*ESC*){nbspace 1}' || compress(put(xu_bin_ * 100, 5.1)) || ');
label cnp_ci='95% CI';
run;

proc datasets lib=work nolist gennum=all;
delete _cnp obsprop;
run;

proc sort data=_base18;
by _byvar1 _cat _rowsrt _rwlabel _trt;
run;

proc sort data=cnpobsprop1;
by _byvar1 _cat _rowsrt _rwlabel _trt;
run;

data _base18;
merge _base18(in=a) cnpobsprop1;
by _byvar1 _cat _rowsrt _rwlabel _trt;

if a;

if compress(_cvalue)="0" then
do;
cnp_ci="NE";
end;
run;

*****
* Specification 5.10: Statistics for Use of antipyretic or pain medication *;
*****
* Specification 5.10.1: Count denominator (N) *;
*****

data _anal19;
set _anal1;
_blcksrt=10;
run;

proc sort data=_anal19;
by _datasrt _byvar1 _blcksrt DENOMFL _trt _cat;
run;

proc sort data=_anal19 out=_catby19(keep=_byvar1) nodupkey;
by _byvar1;
where paramcd eq upcase("Paimedgi");
run;

data _temp19;
set _anal19;
output;
run;

```

```

proc sort data=_temp19 out=_temp919 nodupkey;
  by _datasrt _byvar1 _blcksrt _cat DENOMFL _trt usubjid;
  where paramcd eq upcase("Paimedgi");
run;

proc freq data=_temp919 noprint;
  format DENOMFL;
  tables _datasrt*_byvar1*_blcksrt*_cat * DENOMFL * _trt / sparse norow nocol
    nopercnt out=_pct19(drop=percent);
run;

proc freq data=_pct19 noprint;
  where DENOMFL ne 9999;
  weight count;
  tables _datasrt*_byvar1*_cat * _trt / sparse noprint
    out=_denom19(drop=percent);
run;

data _denomf19;
  length _cat $100;
  _datasrt=1;
  set _catby19(keep=_byvar1);
  * All treatment groups ;
  _trt1=0;
  _trt2=0;
  * _CAT is the subgroup variable ;
  _cat="12-15 Years ";
  output;
  _cat="16-25 Years ";
  output;
run;

proc transpose data=_denom19 out=_denomin19(drop=_name__label_) prefix=_trt;
  by _datasrt _byvar1 _cat;
  var count;
  id _trt;
run;

data _frame19;
  _datasrt=1;
  set _catby19(keep=_byvar1);
  _blcksrt=10;
  length DENOMFL 8;
  length _cat $100;
  _catLabl=" ";
  _trt=1;
  DENOMFL=1;
  _catord=1;
  _subcat=1;
  _cat="12-15 Years ";
  output;
  _subcat=2;
  _cat="16-25 Years ";
  output;

```

```

    _trt=2;
    DENOMFL=1;
    _catord=1;
    _subcat=1;
    _cat="12-15 Years ";
    output;
    _subcat=2;
    _cat="16-25 Years ";
    output;
run;

proc sort data=_frame19;
    by _datasrt _byvar1 _blcksrt _cat DENOMFL _trt;
run;

proc sort data=_pct19;
    by _datasrt _byvar1 _blcksrt _cat DENOMFL _trt;
run;

data _pct19;
    merge _frame19(in=_inframe) _pct19;
    by _datasrt _byvar1 _blcksrt _cat DENOMFL _trt;

    if _inframe;

    if count=. then
        count=0;
run;

proc sort data=_pct19;
    by _datasrt _byvar1 _blcksrt DENOMFL;
run;

data _miss19(keep=_datasrt _byvar1 _blcksrt DENOMFL totcount);
    set _pct19;
    where DENOMFL=9998;
    retain totcount;
    by _datasrt _byvar1 _blcksrt DENOMFL;

    if first.DENOMFL then
        totcount=0;
    totcount=totcount+count;

    if last.DENOMFL;
run;

data _pct19(drop=totcount);
    merge _pct19 _miss19;
    by _datasrt _byvar1 _blcksrt DENOMFL;

    if totcount=0 then
        delete;
run;

```

```

proc sort data=_denomf19;
    by _datasrt _byvar1 _cat;
run;

proc sort data=_denomin19;
    by _datasrt _byvar1 _cat;
run;

data _denomin19;
    merge _denomf19(in=_inframe) _denomin19;
    by _datasrt _byvar1 _cat;

    if _inframe;
    _blcksrt=10;
run;

proc sort data=_pct19;
    by _datasrt _byvar1 _cat;
run;

data _pct19;
    if 0 then
        set _basetemplate;
    merge _denomin19(in=_a) _pct19;
    by _datasrt _byvar1 _cat;

    if _a;
    _varname="DENOMFL ";
    _vrlabel=" ";
    _rwlabel="Use of antipyretic or pain medication(*ESC*){super h} ";

    if DENOMFL=9998 then
        do;
            _rwlabel="Missing ";
            _catord=9998;
        end;
    else if DENOMFL=9999 then
        do;
            _rwlabel="Total ";
            _catord=9999;
        end;

    if _catord=. then
        _catord=9997;
run;

proc sort data=_pct19;
    by _datasrt _byvar1 _blcksrt _catord DENOMFL _trt _cat;
run;

data _base19;
    length _catlab1 $200;
    set _pct19 end=eof;
    by _datasrt _byvar1 _blcksrt _catord DENOMFL _trt _cat;

```

```

retain _rowsrt 0 _rowmax 0;
array _trtcnt(*) _trt1-_trt3;
drop _rowmax _cpct;
length _cpct $100;
_cpct=' ';
_module='mcatstat';

if count > . then
    _cvalue=put(count, 5.);
else
    _cvalue=put(0, 5.);

if length(_cvalue) < 5 then
    do;
        *-----;
        * Put character A0x at right most character to pad text;
        *-----;
        substr(_cvalue, 5, 1)='A0'x;
    end;

if first._byvar1 then
    _rowsrt=0;

if first.DENOMFL then
    do;
        _rowsrt=_rowsrt + 1;
        _rowmax=max(_rowsrt, _rowmax);
    end;
_datatyp='data';
_indent=0;
_dptindt=0;
_vorder=1;
_rowjump=1;

if upcase(_rwlabel)='_NONE_' then
    _rwlabel=' ';
_indent=0;
_dptindt=0;

if _trt=2 +1 then
    _trt=9999;

if eof then
    call symput('_rowsrt', compress(put(_rowmax, 4.)));
_direct="TOP ";
_p=0;

run;

*****
* Specification 5.10.2: Count n and percentage (%) for individual severity    *;
*****

```

data \_anal20;



```

    set _anal2;
    _blcksrt=10;
run;

proc sort data=_anal20;
    by _datasrt _byvar1 _blcksrt AVAL _trt _cat;
run;

proc sort data=_anal20 out=_catby20(keep=_byvar1) nodupkey;
    by _byvar1;
    where paramcd eq upcase("Paimedgi") and ex_none_flg=0 and eventfl="Y";
    ;
run;

data _temp20;
    set _anal20;
    output;
run;

proc sort data=_temp20 out=_temp920 nodupkey;
    by _datasrt _byvar1 _blcksrt _cat AVAL _trt usubjid;
    where paramcd eq upcase("Paimedgi") and ex_none_flg=0 and eventfl="Y";
run;

proc freq data=_temp920 noprint;
    format AVAL;
    tables _datasrt*_byvar1*_blcksrt*_cat * AVAL * _trt / sparse norow nocol
        nopercnt out=_pct20(drop=percent);
run;

proc sort data=_temp20 out=_analcnt20 nodupkey;
    by _datasrt _byvar1 _cat _trt USUBJID;
    where paramcd eq upcase("Paimedgi") and knowvfl eq 'Y';
run;

proc freq data=_analcnt20 noprint;
    tables _datasrt*_byvar1*_cat * _trt / sparse noprint
        out=_denom20(drop=percent);
run;

data _denomf20;
    length _cat $100;
    _datasrt=1;
    set _catby20(keep=_byvar1);
    * All treatment groups ;
    _trt1=0;
    _trt2=0;
    * _CAT is the subgroup variable ;
    _cat="12-15 Years ";
    output;
    _cat="16-25 Years ";
    output;
run;

```

```
proc transpose data=_denom20 out=_denomin20(drop=_name_ _label_) prefix=_trt;
  by _datasrt _byvar1 _cat;
  var count;
  id _trt;
run;
```

```
data _frame20;
  _datasrt=1;
  set _catby20(keep=_byvar1);
  _blcksrt=10;
  length AVAL 8;
  length _cat $100;
  _catLabl=" ";
  _trt=1;
  AVAL=1;
  _catord=1;
  _subcat=1;
  _cat="12-15 Years ";
  output;
  _subcat=2;
  _cat="16-25 Years ";
  output;
  _trt=2;
  AVAL=1;
  _catord=1;
  _subcat=1;
  _cat="12-15 Years ";
  output;
  _subcat=2;
  _cat="16-25 Years ";
  output;
run;
```

```
proc sort data=_frame20;
  by _datasrt _byvar1 _blcksrt _cat AVAL _trt;
run;
```

```
proc sort data=_pct20;
  by _datasrt _byvar1 _blcksrt _cat AVAL _trt;
run;
```

```
data _pct20;
  merge _frame20(in=_inframe) _pct20;
  by _datasrt _byvar1 _blcksrt _cat AVAL _trt;

  if _inframe;

  if count=. then
    count=0;
run;
```

```
proc sort data=_pct20;
```

```

    by _datasrt _byvar1 _blcksrt AVAL;
run;

data _miss20(keep=_datasrt _byvar1 _blcksrt AVAL totcount);
    set _pct20;
    where AVAL=9998;
    retain totcount;
    by _datasrt _byvar1 _blcksrt AVAL;

    if first.AVAL then
        totcount=0;
    totcount=totcount+count;

    if last.AVAL;
run;

data _pct20(drop=totcount);
    merge _pct20 _miss20;
    by _datasrt _byvar1 _blcksrt AVAL;

    if totcount=0 then
        delete;
run;

proc sort data=_denomf20;
    by _datasrt _byvar1 _cat;
run;

proc sort data=_denomin20;
    by _datasrt _byvar1 _cat;
run;

data _denomin20;
    merge _denomf20(in=_inframe) _denomin20;
    by _datasrt _byvar1 _cat;

    if _inframe;
    _blcksrt=10;
run;

proc sort data=_pct20;
    by _datasrt _byvar1 _cat;
run;

data _pct20;
    if 0 then
        set _basetemplate;
    merge _denomin20(in=_a) _pct20;
    by _datasrt _byvar1 _cat;

    if _a;
    _varname="AVAL ";
    _vrlabel=" ";
    _rwlabel="Use of antipyretic or pain medication(*ESC*){super h} ";

```

```

if AVAL=9998 then
  do;
    _rwlabel="Missing ";
    _catord=9998;
  end;
else if AVAL=9999 then
  do;
    _rwlabel="Total ";
    _catord=9999;
  end;

if _catord=. then
  _catord=9997;

run;

proc sort data=_pct20;
  by _datasrt _byvar1 _blcksrt _catord AVAL _trt _cat;
run;

data _base20;
  length _catlabl $200;
  set _pct20 end=eof;
  by _datasrt _byvar1 _blcksrt _catord AVAL _trt _cat;
  retain _rowsrt 1 _rowmax 0;
  array _trtcnt(*) _trt1-_trt3;
  drop _rowmax _cpct;
  length _cpct $100;
  _cpct='';
  _module='mcatstat';

  if count > . then
    _cvalue=put(count, 5.);
  else
    _cvalue=put(0, 5.);
  *-----;
  * Format percent to append to display value in _CVALUE ;
  *-----;

  if _trt ne . then
    do;

      if _trtcnt(_trt) > 0 then
        do;
          percent=count / _trtcnt(_trt) * 100;

          if percent > 0 then
            do;

              if round(percent, 0.1) GE 0.1 then
                _cpct="(*ESC*){nbspspace 1}("||strip(put(percent, 5.1))||")";
              else
                _cpct="(*ESC*){nbspspace 1}{(0.0)";
              _cvalue=trim(_cvalue)||_cpct;
            end;
          else
            _cvalue=trim(_cvalue)||_cpct;
        end;
      else
        _cvalue=trim(_cvalue)||_cpct;
    end;
  end;

```

FDA-CBER-2022-5812-0735819

```

                                end;
        end;
end;

if length(_cvalue) < 13 then
do;
    *-----;
    * Put character A0x at right most character to pad text;
    *-----;
    substr(_cvalue, 13, 1)='A0'x;
end;

if first._byvar1 then
    _rowsrt=1;

if first.AVAL then
do;
    _rowsrt=_rowsrt + 1;
    _rowmax=max(_rowsrt, _rowmax);
end;
_datatyp='data';
_indent=0;
_dptindt=0;
_vorder=1;
_rowjump=1;

if upcase(_rwlabel)='_NONE_' then
    _rwlabel=' ';
_indent=0;
_dptindt=0;

if _trt=2 +1 then
    _trt=9999;

if eof then
    call symput('_rowsrt', compress(put(_rowmax, 4.)));
_direct="TOP ";
_p=2;
run;

data _base20;
set _base20;
length _cvalue2 $30.;
_cvalue2=strip(tranwrd(_cvalue, 'A0'x, ""));
_cvalue21=strip(scan(_cvalue, 1, '('));
_cvalue22=compress(scan(_cvalue, 2, '(', ')'));
run;

data _base19;
set _base19;
drop _trt1 _trt2 count;
run;

proc sort data=_base19;

```

```

    by _datasrt _byvar1 _cat _trt;
run;

data _base19;
    set _base19;

    if _trt=1 then
        do;
            _trt1=input(_cvalue, ??best.);
        end;

    if _trt=2 then
        do;
            _trt2=input(_cvalue, ??best.);
        end;
run;

proc sort data=_base20(keep=_datasrt _trt _cvalue2 _cvalue21 _cvalue22 _cat
    _byvar1 count);
    by _datasrt _byvar1 _cat _trt;
quit;

data _base20;
    merge _base19(in=a) _base20(in=b);
    by _datasrt _byvar1 _cat _trt;

    if a;

    if a and not b then
        do;
            _cvalue2="0";
            _cvalue21="0";
        end;

    if compress(_cvalue2)="0" then
        _cvalue22=put(0, 5.1);

    if compress(_cvalue)="0" then
        do;
            _cvalue2="NA";
            _cvalue21="NA";
            _cvalue22="NA";
        end;
run;

data _base19;
    set _base19;
    delete;
run;

```

```

*****
* Specification 5.10.3: Calculate 95% CI for observed proportion          *;
*****

```

```

data _cnp_tmp_cnp;
  set _base20;

  if count=. then
    count=0;
  indc=1;
  output _cnp;
  indc=2;

  if _trt=1 then
    do;
      count=_trt1 - count;
    end;

  if _trt=2 then
    do;
      count=_trt2 - count;
    end;
  output _cnp;

  if indc=2 and count=0 then
    output _tmp_cnp;
run;

proc sort data=_cnp;
  by _byvar1 _cat _rowsrt _rwlabel _trt;
run;

proc sort nodupkey data=_tmp_cnp(keep=_byvar1 _cat _rowsrt _rwlabel _trt);
  by _byvar1 _cat _rowsrt _rwlabel _trt;
run;

*****
* Call proc freq procedure to calculate CI for observed proportion      *;
*****

proc freq data=_cnp noprint;
  by _byvar1 _cat _rowsrt _rwlabel _trt;
  table indc/binomial alpha=0.05;
  output out=obsprop binomial;
  weight count;
run;

data obsprop;
  merge obsprop _tmp_cnp(in=a);
  by _byvar1 _cat _rowsrt _rwlabel _trt;

  if _bin_=1 and not a then
    do;
      xl_bin_=1 - xu_bin;
      xu_bin_=1 - xl_bin;
    end;
  else
    do;

```

```

        xl_bin_=xl_bin;
        xu_bin_=xu_bin;
    end;
run;

data cnpobsprop1(keep=_byvar1 _cat _rowsrt _rwlabel _trt cnp_ci);
    set obsprop;
    by _byvar1 _cat _rowsrt _rwlabel _trt;
    cnp_ci=(' || compress(put(xl_bin_ * 100, 5.1))
        || '(*ESC*){nbspace 1}' || compress(put(xu_bin_ * 100, 5.1)) || ');
    label cnp_ci='95% CI';
run;

proc datasets lib=work nolist gennum=all;
    delete _cnp obsprop;
run;

```

```

proc sort data=_base20;
    by _byvar1 _cat _rowsrt _rwlabel _trt;
run;

```

```

proc sort data=cnpobsprop1;
    by _byvar1 _cat _rowsrt _rwlabel _trt;
run;

```

```

data _base20;
    merge _base20(in=a) cnpobsprop1;
    by _byvar1 _cat _rowsrt _rwlabel _trt;

    if a;

    if compress(_cvalue)="0" then
        do;
            cnp_ci="NE";
        end;
run;

```

```

*****
* Specification 6
* 1) Generate final report dataset
* 2) Titles and footnotes
* 3) Display: output html file
*****

```

```

data _final;
    set _base1 _base2 _base3 _base4 _base5 _base6 _base7 _base8 _base9 _base10
        _base11 _base12 _base13 _base14 _base15 _base16 _base17 _base18 _base19
        _base20;
run;

proc sort data=_final;
    by _datasrt _byvar1 _blcksrt _rowsrt;
run;

```



```

data _bydata;
    set _bydat1;

    if _byvar1=0 then
        delete;
run;

proc sort data=_bydata;
    by _datasrt _byvar1;
run;

data _final;
    merge _bydata _final(in=_b);
    by _datasrt _byvar1;

    if _b;
run;

*-----;
* Generate treatment header labels and make further modifications ;
*-----;

proc sort data=_final;
    by _trt _subcat;
run;

data _final;
    merge _subGrpData _final(in=_b drop=_colabel);
    by _trt _subcat;

    if _b;
run;

proc sort data=_final;
    by _datasrt _byvar1 _blcksrt _rowsrt;
run;

data _final;
    set _final;
    drop __trt;

    if _trt=9999 then
        __trt=2 + 1;
    else
        __trt=_trt;

    if __trt=. then
        __trt=1;

    if _subcat in (., 9990, 9999) then
        _subcat=2;

    if _subcat < 9990 then

```

```

        _column=_subcat + (__trt - 1) * 2;
else
    _column=_subcat;
    _colabel=translate(trim(_colabel), '^', ' ');

    if _column=9999 then
        _column=2 + 1;
run;

proc sort data=_final out=_final;
    by _datasrt _byvar1 _blcksrt _rowsrt _column;
run;

proc sql;
    create table _final as select *, sum(count) as sum from _final group by
        _blcksrt, _cat, _rowsrt, _rwlabel, _byvar1;
quit;
proc sql noprint;
    create table rspan as select distinct _trt, _column, _subcat, _colabel,
        _byvar1, _bylab1, _vrlabel as _rwlabel, _datasrt, _blcksrt,
        (min(_rowsrt)-0.5) as _rowsrt, _dptindt as _indent, 0 as _dptindt from
        _final(where=(_vrlabel^=' ')) group by _trt, _column, _subcat, _byvar1,
        _datasrt, _blcksrt, _vrlabel;
quit;

data ADCE_S020_SE_sev_ped_SAF;
    length _rvalue $100;
    set _final rspan end=eof;
    _rwindt=sum(_indent, _dptindt);

    if _rwindt <=0 then
        _rvalue=_rwlabel;
    else
        _rvalue=repeat(byte(160), _rwindt-1)||_rwlabel;
    _dummy=1;

    if _trt=. then
        _trt=1;
run;

proc sort data=ADCE_S020_SE_sev_ped_SAF;
    by _datasrt _byvar1 _bylab1 _trt _blcksrt _rowsrt;
run;

data ADCE_S020_SE_sev_ped_SAF;
    set ADCE_S020_SE_sev_ped_SAF;
    _cvalue=left(compress(_cvalue, 'A0'x));
run;

data treat;
    length FMTNAME $8 start 8 label $200;
    fmtname='TREAT';

```

```

do start=1 to 2 + ("N"="Y");
    label=symget('_TRTLB'|| compress(put(start, 4.)));
    label=trim(label);
    output;
end;
run;

proc sql;
    create table subcat as select distinct 'SUBCAT' as FMTNAME length=8 , _subcat
        as start, tranwrd(_colabel, '^', ' ') as label from
        ADCE_S020_SE_SEV_PED_SAF order by fmtname, start;
quit;
proc format cntlin=subcat;

proc sql noprint;
    select distinct start, label, count(distinct start) into :start1,
        :_trlb11 - :_trlb199, :maxtrt from treat where start ne 9999 order by start;
quit;

*-----;
* titles and footnotes ;
*-----;

options orientation=LANDSCAPE papersize="LETTER";
ods escapechar="~";

title1 "Systemic Events, by Maximum Severity, Within 7 Days After Each Dose (*ESC*){Unicode 2013}";
title2 "Subjects 12 Through 15 and 16 Through 25 Years of Age (Reactogenicity Subset) (*ESC*){Unicode 2013}
Safety Population";

footnote1 "Note: Events and use of antipyretic or pain medication were collected in the electronic diary (e-diary) from
Day 1 through Day 7 after each dose. Grade 4 events were classified by the investigator or medically qualified person. ";
footnote2 "Note: Subject C4591001 1077 10771278 (13 years of age) experienced systemic events, including a
temperature of 40.4(*ESC*){Unicode 00B0}C, on the day of Dose 2. Since these events were recorded as adverse
events and not in the e-diary, they do not appear in this table.";
footnote3 "a.(*ESC*){nbspspace 5}N = number of subjects reporting at least 1 yes or no response for the specified event
after the specified dose.";
footnote4 "b.(*ESC*){nbspspace 5}n = Number of subjects with the specified characteristic. ";
footnote5 "c.(*ESC*){nbspspace 5}Exact 2-sided CI based on the Clopper and Pearson method.";
footnote6 "d.(*ESC*){nbspspace 5}Mild: does not interfere with activity; moderate: some interference with activity;
severe: prevents daily activity; Grade 4: emergency room visit or hospitalization for severe fatigue, severe headache,
severe chills, severe muscle pain, or severe joint pain. ";
footnote7 "e.(*ESC*){nbspspace 5}Mild: 1 to 2 times in 24 hours; moderate: >2 times in 24 hours; severe: requires
intravenous hydration; Grade 4: emergency room visit or hospitalization for severe vomiting.";
footnote8 "f.(*ESC*){nbspspace 5}Mild: 2 to 3 loose stools in 24 hours; moderate: 4 to 5 loose stools in 24 hours; severe:
6 or more loose stools in 24 hours; Grade 4: emergency room visit or hospitalization for severe diarrhea. ";
footnote9 "g.(*ESC*){nbspspace 5}Any systemic event: any fever (*ESC*){unicode 2265}38.0(*ESC*){Unicode
00B0}C, any fatigue, any vomiting, any chills, any diarrhea, any headache, any new or worsened muscle pain, or any
new or worsened joint pain.";
footnote10 "h.(*ESC*){nbspspace 5}Severity was not collected for use of antipyretic or pain medication. ";

*-----;
* Output html file;
*-----;

```

```
ods html file="&outtable.";
```

```
data report;  
  set ADCE_S020_SE_sev_ped_SAF;  
  
  if _trt=9999 then  
    _trt=2 +1;  
  _bylab1=tranwrd(_bylab1, "|", '036e'x);  
  _rvalue=tranwrd(_rvalue, "|", '036e'x);  
run;
```

```
proc sort data=report;  
  by _datasrt _byvar1 _bylab1 _blcksrt _rowsrt _rvalue _subcat _trt;  
run;
```

```
data data_1 (keep=_datasrt _byvar1 _bylab1 _blcksrt _rowsrt _rvalue _subcat  
  COL:);  
  set report;  
  where _trt=1;  
  rename _cvalue=COL11 _cvalue2=COL12 cnp_ci=COL13;  
run;
```

```
data data_2 (keep=_datasrt _byvar1 _bylab1 _blcksrt _rowsrt _rvalue _subcat  
  COL:);  
  set report;  
  where _trt=2;  
  rename _cvalue=COL21 _cvalue2=COL22 cnp_ci=COL23;  
run;
```

```
proc sort data=report out=extradata (keep=_datasrt _byvar1 _bylab1 _blcksrt  
  _rowsrt _rvalue _subcat) nodupkey;  
  by _datasrt _byvar1 _bylab1 _blcksrt _rowsrt _rvalue _subcat;  
run;
```

```
data report;  
  merge data_1 data_2 extradata;  
  by _datasrt _byvar1 _bylab1 _blcksrt _rowsrt _rvalue _subcat;  
run;
```

```
data report;  
  set report;  
  _fixvar=1;  
  _fix2var=1;  
  _dummy=1;  
run;
```

```
proc sort data=report out=outdata1;  
  by _datasrt _byvar1 _bylab1 _blcksrt _rowsrt _rvalue _subcat;  
run;
```

```

*-----;
* proc report statements ;
*-----;

proc report data=outdata1 nowd list missing contents="" split="|" spanrows style(report)={} style(header)={}
style(column)={};
    column _fixvar _fix2var _datasrt _byvar1 _bylab1 _blksrt _rowsrt (" " " " " _rvalue) (("Vaccine Group (as
Administered)~{line}"
    ("&_trbl1." _subcat, (COL11 COL12 COL13)) ("&_trbl2." _subcat, (COL21 COL22 COL23))) ) _dummy;
    define _fixvar / group noprint;
    define _fix2var / group noprint;
    define _subcat / across order=internal ' ' format=subcat. style(header)={just=center} center;
    define _byvar1 / group order=internal noprint;
    define _bylab1 / group "Dose" style(column)={just=left} style(header)={just=left} left;
    define _datasrt / group order=internal noprint;
    define _blksrt / group order=internal noprint;
    define _rowsrt / group order=internal noprint;
    define _rvalue / group "Systemic Event" order=data style(column)={just=left} style(header)={just=left} left;
    define COL11 / group nozero "N(*ESC*){super a}" style(column)={leftmargin=12px} style(header)=
{jst=center} center;
    define COL12 / group nozero "n(*ESC*){super b} (%)" style(column)={leftmargin=12px} style(header)=
{jst=center} center;
    define COL13 / group nozero "(95% CI(*ESC*){super c})" style(column)={leftmargin=12px} style(header)=
{jst=center} center;
    define COL21 / group nozero "N(*ESC*){super a}" style(column)={leftmargin=12px} style(header)=
{jst=center} center;
    define COL22 / group nozero "n(*ESC*){super b} (%)" style(column)={leftmargin=12px} style(header)=
{jst=center} center;
    define COL23 / group nozero "(95% CI(*ESC*){super c})" style(column)={leftmargin=12px} style(header)=
{jst=center} center;
    define _dummy / sum noprint;
    break before _fixvar / contents="" page;
    compute before _fix2var;
        line @1 " ~n ";
    endcomp;
    compute after _blksrt;
        line " ~n ";
    endcomp;
run;

ods html close;
proc printto;
run;

```