

/*****

** STUDY ID : 000000106343

**

** PROGRAM NAME : t_ecg.sas

**

** DATE : 15May2015

**

** PROGRAMMER : cvn_aramasah

**

** PURPOSE : QC the table Summary of ECG results - safety population
(t_15_2_6_21)

**

** INPUT DATA : ADAM.ADSL, ADAM.ADEG

**

** OUTPUT DATA :

**

** SAS MACROS USED :

**

** MODIFICATIONS : DATE : MODIFIED BY : NOTES :

**

**

** PROGRAMMED USING SAS VERSION 9.3 **

** COPYRIGHT (C) 2015 BY COVANCE, PRINCETON NJ - USA - ALL RIGHTS RESERVED **

-----/

```

%m_printto;

options notes nosource replace;

proc datasets lib=work nolist memtype=data kill; quit;

options notes source source2 nofullstimer validvarname=upcase missing=' ';

ods _all_ close;

ods listing;

*=====;

* START OF PROGRAM CODE                                ;

*=====;

/* Standard - just change the number to match the listing you're working on. Also change the letters in
the*/

/* bracket, eg ccb = current cigarette brands. Make sure to do this at the top of the code too. */

%let tflno=T_15_02_06_21;

%let TFLprg=t_ecg.sas;

/* Standard - leave this */

%let TFL_Part=%scan(&_SASPROGRAMFILE,-3,%str(/));

/* Standard - leave this */

data _null_;

    tmp("&TFL_Part";

        if tmp not in ("dev" "qc") then call symput("TFL_Part", "prod");

```

```
call symput('TFLpath', compress("&_SASPROGRAMFILE", ""));  
run;
```

```
*****,
```

```
* read in data ;
```

```
*****,
```

```
/*Use ADSL to get N numbers for column headers*/
```

```
data adsl;
```

```
    set adam.adsl;
```

```
    if trt01a='THSm2.2' then trt01an=1;
```

```
    if trt01a='mCC' then trt01an=2;
```

```
    if trt01a='SA' then trt01an=3;
```

```
    where safaf1 = 'Y';
```

```
    if missing(trt01an) then delete;
```

```
    if index(trt01a,'Exposed') then delete;
```

```
    output;
```

```
    trt01an=99;
```

```
    trt01a='Overall Safety';
```

```
    output;
```

```
run;
```

```
proc freq data=adsl noprint;
```

```

table trt01an*trt01a/ out =tot(drop=percent);

run;

data dumtrts; /*Use this to output any columns for which N=0*/

    attrib trt01a length =$40.

                trt01an length=8.;

    trt01an=1;

    trt01a='THSm2.2';

    output;

    trt01an=2;

    trt01a='mCC';

    output;

    trt01an=3;

    trt01a='SA';

    output;

    trt01an=99;

    trt01a='Overall Safety';

    output;

run;

data tot2;

    merge tot(in=a) dumtrts(in=b);

    by trt01an trt01a;

    if a or b;

    if b and not a then count=0;

```

```

        call symput('trt' || compress(put(trt01an,best.)), compress(put(count,best.)));

run;

/*Bring in appropriate data from ADEG*/

data adeg;

    set adam.adeg;

    if trta='THSm2.2' then trtan=1;

    if trta='mCC' then trtan=2;

    if trta='SA' then trtan=3;

    if ablfl='Y' then anl01fl='Y';

    where safabl = 'Y';* and anl01fl='Y';

/* 7) start JMH 07Oct2014 */

    IF ABLFL='Y' THEN DO;

        AVISITN=1;

        AVISIT='Baseline';

    END;

    IF AVISIT NE 'Baseline' AND AVISITN LE 1 THEN DELETE;

/* 7) end JMH 07Oct2014 */

    if missing(trtan) then delete;

    if index(trta,'Exposed') then delete;

    output;

    trtan=99;

    trta='Overall Safety';

    output;

```

```

        if upcase(avisit) ne 'BASELINE' and asper not in (2 3 4) then delete;

run;

/* for baseline */

data ABLFL;

    set adam.adeg;

    where ablfl='Y' and safabl='Y';

    basedate=egdtc;

    if ablfl='Y' then do; avisitn=100; avisit='Baseline'; end;

    if trta='THSm2.2' then trtan=1;

    if trta='mCC' then trtan=2;

    if trta='SA' then trtan=3;

    output;

    trtan=99;

    trta='Overall Safety';

    output;

    keep usubjid basedate paramcd trta trtan ;

run;


proc sort data=adeg;

by usubjid paramcd trtan trta;

proc sort data=ablfl;

by usubjid paramcd trtan trta;

run;

```

```
data adeg_adeq;  
merge adeg ablfl;  
by usubjid paramcd trtan trta;  
run;
```

```
data adeg_;  
set adeg_adeq;  
    if index((avisit), 'Unscheduled') gt 0 then delete;  
    if anl01fl='Y' and egdtc ge basedate;  
run;  
/*INTP only as this code will bring out the class variables. Other params will be dealt with later*/
```

```
proc freq data=adeq_(where=(paramcd='INTP')) noprint;  
    table trtan*trta*avisitn*avisit*aval*avalc*paramn*param*EGCLSIG / out =intp1(drop=percent);  
/* 7) JMH 07Oct2014 */  
run;
```

```
proc freq data=adeq_(where=(paramcd='INTP')) noprint;  
    table trtan*trta*avisitn*avisit / out =intp1_n(drop=percent); /* 7) JMH 07Oct2014 */  
run;
```

```
data intp1;  
set intp1;  
order=avisitn;  
run;
```

```
data intp1_n;  
set intp1_n;  
order=avisitn-0.01;  
avalc='n';  
paramn=7;  
param='Interpretation';  
run;
```

```
data intp1;  
set intp1 intp1_n;  
run;
```

```
proc sort data=intp1;by trtan order;run;
```

```
data dumtrts2; /*Use this to output any columns for which N=0*/  
    attrib trta length =$40.  
            trtan length=8.;  
    trtan=1;  
    trta='THSm2.2';  
    output;  
    trtan=2;  
    trta='mCC';
```



```
output;

trtan=3;

trta='SA';

output;

trtan=99;

trta='Overall Safety';

output;

run;
```

```
data tot3;

    set tot2;

    trta=trt01a;

    trtan=trt01an;

    drop trt01an trt01a;

run;
```

```
data intp2;

    merge intp1(in=a drop=paramn param) dumtrts2(in=b) tot3(rename=(count=total));

    by trtan trta;

    attrib param length=$40.

           paramn length=8.;

    if a or b;

/* 7) START JMH 07Oct2014 */

/*    if b and not a then do;*/

/*    count = 0;*/
```

```

/*      avisitn=1;*/

/*      avisit='Screening';*/

/*      avalc='Normal';*/

/*      end;*/

/* 7) END JMH 07Oct2014 */

/*      if total ne 0 then percent=count/total*100;*/

/*      else percent=0; */

      paramn=0;

      param='Clinical Relevance Assessment';

run;

proc sort data=intp2 nodupkey out=trtvis(keep=trtan trta avisitn avisit avalc paramn param);

      by trtan trta avisitn avisit;

run;

data dumrows;

set trtvis;

      avalc='Normal';

      output;

/* 7) start JMH 07Oct2014 */

/*      avalc='Abnormal, CNR';*/

      AVALC='Abnormal';

      EGCLSIG='NCS';

      output;

```

```

/*      avalc='Abnormal, CR';*/

      AVALC='Abnormal';

      EGCLSIG='CS';

/* 7) end JMH 07Oct2014 */

      output;

run;


proc sort data=dumrows;

      by trtan trta avisitn avisit avalc EGCLSIG paramn param; /* 7) JMH 07Oct2014 */

run;


proc sort data=intp2;

      by trtan trta avisitn avisit avalc EGCLSIG paramn param; /* 7) JMH 07Oct2014 */

run;


data intp3;

      merge intp2(in=a) dumrows(in=b);

      by trtan trta avisitn avisit avalc EGCLSIG paramn param; /* 7) JMH 07Oct2014 */

      if a or b;

      attrib statval length=$100.;

      if a then data=1;

      if b then dummy=1;

      if b and not a then do;

              count=0;

```

```

end;

attrib paramc length = $100.

stat length = $100.;

paramc=strip(param);

if trtan=97 then do;

    paramc=strip('Clinical Relevance Assessment');

    paramn=0;

end;

ord = avisitn;

if avalc='n' then do;

    stat='n';

    statord=1;

end;

else if avalc='Normal' then do;

    stat='Normal';

    statord=2;

end;

else if avalc='Abnormal'/*, CNR'*/ AND EGCLSIG='NCS' then do; /* 7) JMH 07Oct2014 */

    stat=/*'Abnormal non-clinically relevant - n (%)'*/'Abnormal NCS'; /* 7) JMH

07Oct2014 */

    statord=3;

end;

else if avalc='Abnormal'/*, CR'*/ AND EGCLSIG='CS' then do; /* 7) JMH 07Oct2014 */

    stat=/*'Abnormal clinically relevant - n (%)'*/'Abnormal CS'; /* 7) JMH

07Oct2014 */

```

```

            statord=4;

        end;

        statval=strip(put(count,best.)) ;

run;

proc sort data=intp3;

        by paramn paramc ord avisit avisitn statord stat;

run;

proc transpose data=intp3 out=intp4a(where=(stat ne 'DUMMY')) prefix=t;

        by paramn paramc ord avisit avisitn statord stat;

        var statval;

        id trtan;

        idlabel trta;

run;

data intp_perc;

set intp4a;

where stat='n';

t1_=input(t1,best.);

t2_=input(t2,best.);

t3_=input(t3,best.);

t99_=input(t99,best.);

keep avisitn t1_ t2_ t3_ t99_;

```

```
run;
```

```
proc sort data=intp4a;by avisitn;proc sort data=intp_perc;by avisitn;run;
```

```
data intp4;
```

```
merge intp4a intp_perc;
```

```
by avisitn;
```

```
    if t1="" then t1='0';
```

```
    if t2="" then t2='0';
```

```
    if t3="" then t3='0';
```

```
    if t99="" then t99='0';
```

```
if stat ne 'n' then do;
```

```
if compress(t1) ne '0' then t1=strip(t1)||'('|| strip(put(((input(t1,best.))/t1_*100),5.1))||'%');
```

```
if compress(t2) ne '0' then t2=strip(t2)||'('|| strip(put(((input(t2,best.))/t2_*100),5.1))||'%');
```

```
if compress(t3) ne '0' then t3=strip(t3)||'('|| strip(put(((input(t3,best.))/t3_*100),5.1))||'%');
```

```
if compress(t99) ne '0' then t99=strip(t99)||'('|| strip(put(((input(t99,best.))/t99_*100),5.1))||'%');
```

```
end;
```

```
drop t1_ t2_ t3_ t99_;
```

```
run;
```

```
/*End of INTP, will set on with rest of data later*/
```

```
/*Now this will create the stats for all params except INTP*/
```

```
data adeg_orig; /*This is for the actual values so aval will be used as the analysis variable*/
```

```

set adeg_(where=(paramcd ne 'INTP'));

if avisitn=1 then ord=1;

else if avisitn=101 then ord=1;/*Day 1*/

else if avisitn=102 then ord=2;/*Day 2*/

else if avisitn=103 then ord=3;/*Day 3*/

else if avisitn=104 then ord=4;/*Day 4*/

else if avisitn=105 then ord=5;/*Day 5*/

else if avisitn=106 then ord=6;/*Discharge confinement*/

else if avisitn=121 then ord=7;/*Day 21*/

else if avisitn=130 then ord=8;/*Day 30*/

else if avisitn=160 then ord=10;/*Day 60*/

else if avisitn=191 then ord=12;/*Discharge ambulatory*/

/* else put "WA" "RNING: Unexpected value for avisitn: " avisitn= ;*/

statval=aval;

run;

```

```

data adeg_chg; /*This is for the changes from baseline so chg will be used as the analysis variable*/

```

```

*set adeg(where=(avisitn in(106) and paramcd ne 'INTP')); /*Only keep days after baseline*/

```

```

*if avisitn=106 then ord=6;/*Change from Baeline to Day 6*/

```

```

set adeg_(where=(avisitn gt 1 and paramcd ne 'INTP')); /*Only keep days after baseline*/

```

```

if avisitn=101 then ord=1;/*Change from Day 1*/

```

```

else if avisitn=102 then ord=2;/*Change from Day 2*/

```

```

else if avisitn=103 then ord=3;/*Change from Day 3*/

```

```

else if avisitn=104 then ord=4;/*Change from Day 4*/
else if avisitn=105 then ord=5;/*Change from Day 5*/
else if avisitn=106 then ord=6;/*Change from Discharge confinement*/
else if avisitn=121 then ord=7;/*Change from Day 21*/
else if avisitn=130 then ord=8;/*Change from Day 30*/
else if avisitn=160 then ord=10;/*Change from Day 60*/
else if avisitn=191 then ord=12;/*Change from Discharge ambulatory*/
else put "WA" "RNING: Unexpected value for avisitn: " avisitn= ;

statval=chg;

run;

/*Transpose for raw values*/

proc sort data=adeg_orig;

    by trtan trta paramn ord param avalu avisit avisitn;

run;

proc univariate data=adeg_orig noprint;

    var statval;

    by trtan trta paramn ord param avalu avisit avisitn;

    output out=results01_orig n=n1o mean=mean1o std=std1o median=med1o min=min1o max=max1o;

run;

/*Transpose for change from baseline values*/

proc sort data=adeg_chg;

    by trtan trta paramn ord param avalu avisit avisitn;

```



```
run;
```

```
proc univariate data=adeg_chg noprint;
```

```
var statval;
```

```
by trtan trta paramn ord param avalu avisit avisitn;
```

```
output out=results01_chg n=n1c mean=mean1c std=std1c median=med1c min=min1c max=max1c;
```

```
run;
```

```
data results01;
```

```
merge results01_orig results01_chg;
```

```
by trtan trta paramn ord param avalu avisit avisitn;
```

```
run;
```

```
data results02;
```

```
set results01;
```

```
attrib meano length=$10.
```

```
        mino length=$10.
```

```
        no   length=$10.
```

```
        mediano length=$10.
```

```
        stdo length=$10.;
```

```
/*        maxo length=$10.;*/
```

```
        if not missing(n1o) then no = left(compress(put(n1o,8.))); else no='0';
```

```

        if not missing(med1o) then mediano = left(compress(put(med1o,8.1)));
        if not missing(mean1o) then meano = left(compress(put(mean1o,8.1)));
        if not missing(std1o) then stdo =
'(' || left(compress(put(0.01*ceil(std1o/0.01),8.2))) || ')';
        if not missing(min1o) and not missing (max1o) then mino =
left(compress(put(min1o,8.))) || ', ' || left(compress(put(max1o,8.)));
/*          if not missing(max1o) then maxo = left(compress(put(max1o,8.))); */

```

attrib meanc length=\$10.

```

        minc length=$10.
        nc   length=$10.
        medianc length=$10.
        stdc length=$10.;
/*          maxc length=$10.; */

```

```

        if not missing(n1c) then nc= left(compress(put(n1c,8.))); else nc='0';
        if not missing(med1c) then medianc = left(compress(put(med1c,8.1)));
        if not missing(mean1c) then meanc = left(compress(put(mean1c,8.1)));
        if not missing(std1c) then stdc =
'(' || left(compress(put(0.01*ceil(std1c/0.01),8.2))) || ')';
        if not missing(min1c) and not missing (max1c) then minc =
left(compress(put(min1c,8.))) || ', ' || left(compress(put(max1c,8.)));
/*          if not missing(max1c) then maxc = left(compress(put(max1c,8.))); */

```

drop n1o mean1o std1o med1o min1o max1o n1c mean1c std1c med1c min1c max1c;

run;

```

data results03; /*Create text as required in output*/

    set results02;

    attrib paramc length = $100.

                                visit length = $100.;

    if ord=1 then visit=avisit;

    else if ord=2 then visit=avisit;

    else put "WARN" "ING unexpected ord value" ord=;

paramc=strip(param);

    /*This bit of code just populates the variables of dummy columns to avoid
problems with the transpose*/

    if missing(ord) and missing(paramn) then do;

        ord=1;

        visit='Screening';

        paramn=1;

        paramc='Summary (Mean) Heart Rate (BEATS/MIN)';

    end;

run;

proc sort data=results03;

by paramn paramc ord avisit avisitn;

run;

```

```
proc transpose data=results03 out=results04_orig1 prefix=o name=varname;

  by paramn paramc ord avisit avisitn;

  var no meano stdo mediano mino,* maxo;

  id trtan;

  idlabel trta;

run;
```

```
data results04_orig;

  set results04_orig1;

  varname=tranwrd(varname,'O','C');

run;
```

```
proc transpose data=results03 out=results04_chg prefix=c name=varname;

  by paramn paramc ord avisit avisitn;

  var nc meanc stdc medianc minc,* maxc;

  id trtan;

  idlabel trta;

run;
```

```
proc sort data=results04_orig;

  by paramn paramc ord avisit avisitn varname;

run;
```

```
proc sort data=results04_chg;

    by paramn paramc ord avisit avisitn varname;

run;
```

```
data results04;

    merge results04_orig results04_chg;

    by paramn paramc ord avisit avisitn varname;

run;
```

```
data results05;

    set results04;

    attrib stat length = $100.;

    if upcase(varname)='NC' then do; statord=1; stat='n'; end;

    else if upcase(varname)='MEANC' then do; statord=2; stat='Mean'; end;

    else if upcase(varname)='STDC' then do; statord=3; stat='(SD)'; end;

    else if upcase(varname)='MEDIANC' then do; statord=4; stat='Median'; end;

    else if upcase(varname)='MINC' then do; statord=5; stat='Min, Max'; end;

/*    else if upcase(varname)='MAXC' then do; statord=6; stat='Max'; end;*/

    drop varname;

run;

data results06;
```

```

set results05;

if paramc ne 'Interpretation' then do;
    if stat='n' then do;
        if missing(o1) then o1='0';
        if missing(o2) then o2='0';
        if missing(o3) then o3='0';
        if missing(o99) then o99='0';
    end;

    if stat='n' and ord=2 then do;
        if missing(o1) then o1='0';
        if missing(o2) then o2='0';
        if missing(o3) then o3='0';
        if missing(o99) then o99='0';
        if missing(c1) then c1='0';
        if missing(c2) then c2='0';
        if missing(c3) then c3='0';
        if missing(c99) then c99='0';
    end;
end;

run;

/*Now combine the stats with the classification results*/

proc sort data=results06;
    by paramn paramc ord statord;

```

```
run;
```

```
data allresults;
```

```
    set results06 intp4(in=a);
```

```
    by paramn paramc ord statord;
```

```
    if a then do;
```

```
        if missing(t1) then t1='0';
```

```
        if missing(t2) then t2='0';
```

```
        if missing(t3) then t3='0';
```

```
        if missing(t99) then t99='0';
```

```
    end;
```

```
/* 1) start JMH 15Jul2014 */
```

```
/*      IF PARAMC='Heart Rate (Beats/min)' THEN PARAMC='Heart rate (beats/min)';*/
```

```
/*      ELSE IF PARAMC='PR Duration (msec)' THEN PARAMC='PR duration (msec)';*/
```

```
/*      ELSE IF PARAMC='QT Duration (msec)' THEN PARAMC='QT duration (msec)';*/
```

```
/*      ELSE IF PARAMC="QTcB - Bazett's Correction Formula (msec)" THEN PARAMC="QTcB - Bazett's  
correction formula (msec)";*/
```

```
/*      ELSE IF PARAMC="QTcF - Fridericia's Correction Formula (msec)" THEN PARAMC="QTcF -  
Fridericia's correction formula (msec)";*/
```

```
/*      ELSE IF PARAMC='QRS Duration (msec)' THEN PARAMC='QRS duration (msec)';*/
```

```
/*ELSE */IF PARAMC='Clinical Relevance Assessment' THEN DO; PARAMC='Interpretation';  
PARAMN=0; END;
```

```
/*      ELSE PUT "WA" "RNING: Check parameter text is sentence case " PARAMC= ;*/
```

```
/* 1) end JMH 15Jul2014 */
```

```
/*IF INDEX(VISIT,'Screening') THEN VISIT=TRANWRD(VISIT,'Screening','Baseline');*/ /* 2) JMH  
15Jul2014 */ /* 7) JMH 07Oct2014 */
```

```
flag=1;
```

```
run;
```

```
data labels;
```

```
set allresults;
```

```
attrib          or1 or2 or3 or99 label = "Raw value" length=$20.
```

```
                c1 c2 c3 c99 label = "Change"
```

```
                paramc label='Parameter (units)'
```

```
                avisit label='Study Day'
```

```
                stat label='Statistic';
```

```
if paramn = 0 then do; /* 1) JMH 15Jul2014 */
```

```
    or1 = t1;
```

```
    or2 = t2;
```

```
    or3 = t3;
```

```
    or99 = t99;
```

```
    if c1="" then c1='0';
```

```
    if c2="" then c2='0';
```

```
    if c3="" then c3='0';
```

```
    if c99="" then c99='0';
```

```
end;
```

```
if o1 ne "" then do;
```

```
    or1=o1;
```



```

        or2=o2;

        or3=o3;

        or99=o99;

    end;

        if avisit='Baseline' then do;

            c1="";

            c2="";

            c3="";

            c99="";

            end;

/*      avisit=tranwrd(avisit,'/' '/' );*/

            drop o1 o2 o3 o99;

        if paramc='Interpretation' then do;

            c1="";

            c2="";

            c3="";

            c99="";

            end;

run;

data labels;

set labels;

/*if paramn=0 then do;*/

if avisitn in (1 106) then new_page=1;

```

```
if avisitn in (130 160) then new_page=2;
```

```
if avisitn=191 then new_page=3;
```

```
/*end;*/
```

```
/*else do;*/
```

```
/*if avisitn in (1 106) then new_page=1;*/
```

```
/*if avisitn in (*/
```

```
/*end;*/
```

```
run;
```

```
proc sort data=labels out=page (keep= paramn/* avisitn*/ new_page) nodupkey;
```

```
by paramn /*avisitn*/ new_page;
```

```
run;
```

```
data page;
```

```
set page;
```

```
page=_n_;
```

```
run;
```

```
data paging;
```

```
merge labels page;
```

```
by paramn /*avisitn*/ new_page;
```

```
call symput("page",compress(put(page,best.)));
```

```

run;

/*proc sort data=allresults out=extraline (keep=paramc avisit) nodupkey;*/

/*by paramc avisit;*/

/*run;*/

/**/

/*data extraline;*/

/*set extraline;*/

/*if paramc='Heart Rate (Beats/min)' then paramn=1;*/

/*if paramc='Interpretation' then paramn=0;*/

/*if paramc='PR Duration (msec)' then paramn=2;*/

/*if paramc='QRS Duration (msec)' then paramn=6;*/

/*if paramc='QT Duration (msec)' then paramn=3;*/

/*if paramc="QTcB - Bazett's Correction Formula (msec)" then paramn=4;*/

/*if paramc="QTcF - Fridericia's Correction Formula (msec)" then paramn=5;*/

/**/

/*if paramc in ('Heart Rate (Beats/min)' 'Interpretation' 'PR Duration (msec)' 'QRS Duration (msec)' 'QT
Duration (msec)' "QTcB - Bazett's Correction Formula (msec)" "QTcF - Fridericia's Correction Formula
(msec)") and avisit='Baseline' then ord=0.1;*/

/*if paramc in ('Heart Rate (Beats/min)' 'PR Duration (msec)' 'QRS Duration (msec)' 'QT Duration (msec)'
"QTcB - Bazett's Correction Formula (msec)" "QTcF - Fridericia's Correction Formula (msec)") and
avisit='Day 30' then ord=7.9;*/

/*if paramc in ('Heart Rate (Beats/min)' 'PR Duration (msec)' 'QRS Duration (msec)' 'QT Duration (msec)'
"QTcB - Bazett's Correction Formula (msec)" "QTcF - Fridericia's Correction Formula (msec)") and
avisit='Day 6/Discharge Confinement' then ord=5.9;*/

/*if paramc in ('Heart Rate (Beats/min)' 'PR Duration (msec)' 'QRS Duration (msec)' 'QT Duration (msec)'
"QTcB - Bazett's Correction Formula (msec)" "QTcF - Fridericia's Correction Formula (msec)") and
avisit='Day 60' then ord=9.9;*/

```

```
/*if paramc in ('Heart Rate (Beats/min)' 'PR Duration (msec)' 'QRS Duration (msec)' 'QT Duration (msec)'  
"QTcB - Bazett's Correction Formula (msec)" "QTcF - Fridericia's Correction Formula (msec)") and  
avisit='Day 91/Discharge Ambulatory' then ord=11.9;*/
```

```
/**/
```

```
/*if paramc='Interpretation' and avisit='Baseline' then page_=1;*/
```

```
/*if paramc='Interpretation' and avisit='Day 6/Discharge Confinement' then do; ord=105.9; page_=2;  
end;*/
```

```
/*if paramc='Interpretation' and avisit='Day 30' then do; ord=129.9; page_=3; end;*/
```

```
/*if paramc='Interpretation' and avisit='Day 60' then do; ord=159.9; page_=4; end;*/
```

```
/*if paramc='Interpretation' and avisit='Day 91/Discharge Ambulatory' then do; ord=190.9; page_=5;  
end;*/
```

```
/*run;*/
```

```
/*data labels_1;*/
```

```
/*set labels extraline;*/
```

```
/*run;*/
```

```
/**/
```

```
/*proc sort data=labels_1;*/
```

```
/*by paramn ord;*/
```

```
/*run;*/
```

```
/**/
```

```
/*proc sort data=labels_1;*/
```

```
/*      by paramn ord statord;*/
```

```
/*run;*/
```

```
/*data paging;*/
```

```
/* set labels_1;*/
```

```

/*      by paramn ord statord;*/

/**/

/*  flag=1;*/

/*if paramc='Interpretation' then do;*/

/*if avisit='Baseline' then page_=1;*/

/*if avisit='Day 6/Discharge Confinement' then page_=2;*/

/*if avisit='Day 30' then page_=3;*/

/*if avisit='Day 60' then page_=4;*/

/*if avisit='Day 91/Discharge Ambulatory' then page_=5;*/

/*c1=";*/

/*c2=";*/

/*c3=";*/

/*c99=";*/

/*end;*/

/*else if first.ord and ln ge 6 then ln=1; /*Amend to look presentable, and avoid page overflows*/

/*  else ln+1;*/

/*  if ln=1 then page+1;*/

/*  call symput("page",compress(put(page,best)));*/

/*      if ord in (105.9 129.9 159.9 190.9 0.1 7.9 5.9 9.9 11.9) then avisit=";*/

/*      if page_ ne . then page=page_;          */

/**/

/*run;*/;

```

```

proc sort data=paging;

```

```
by paramn;
```

```
run;
```

```
data lastpage_1;
```

```
set paging;
```

```
by paramn;
```

```
if last.paramn;
```

```
run;
```

```
proc sort data=lastpage_1;
```

```
by paramn ord;
```

```
run;
```

```
data lastpage;
```

```
set lastpage_1;
```

```
by paramn ord;
```

```
if last.ord then call symput('maxpage', trim(left(put(page,best.))));
```

```
run;
```

```
proc sql noprint;
```

```
create table tflds.&tfino as
```

```
select paramc, avisit, stat, or1 as THS_raw, c1 as ths_chg, or2 as mcc_raw, c2 as mcc_chg, or3 as sa_raw,  
c3 as sa_chg, or99 as overall_raw, c99 as overall_chg, paramn, page
```

```
from paging
```

```
order by paramn, ord, statord;
```

```
quit;
```

```
/* Standard - leave this */
```

```
options number nodate orientation=landscape /*papersize=&p_pgsize*/ missing=' ';
```

```
ods escapechar='$';
```

```
%let linetop = \brdrt\brdrs\brdrw30; * needs to be 1.5pt so calculated in twips (1/20 pt) ;
```

```
%let linebot = \brdrb\brdrs\brdrw30;
```

```
/* Standard - macro for paging */
```

```
%macro outrtf(blankn=, halfblnk=);
```

```
%if &halfblnk=N %then %let halfblnk=;
```

```
%else %if &halfblnk=Y %then %let halfblnk=\~;
```

```
ods path stdlib.t106343 (read) ;
```

```
ods results off;
```

```
ods rtf toc_data file="/cvn/projects/prj/data/000000106343/TFL/dev/Tables/&tflno..rtf" style=t106343  
startpage=yes headery=1440 footery=1440 ;
```

```
ods noproctitle;
```

```
%do i=1 %to &page;
```

```
title ;
```

```

footnote;

%let wd=0;


%let npage=%eval(&i);


ods proclabel = ' ';


data comp;

    set paging end=eof;

        where page=&i;

/*          call symput('paramc',paramc);*/

        _firtitl="Table 15.2.6.21 Summary of ECG Measurements - Safety Population";

        _secndtitl="Safety Time Period: Randomized Period";

        _upcas=(length("Path: &TFLpath.")-
length(compress("Path:&TFLpath.",'ABCDEFGHIJKLMNOPQRSTUVWXYZ')))/2;

        len=&blankn.-length("(Page &npage of &maxpage)");

        if eof then do;

/*          call symput('paramc',strip(paramc));*/

            call symput('par',put(paramn,best.));

            call symput('_FSRTITL', trim(left(_firtitl)));

            call symput('_SECTITL', trim(left(_secndtitl)));

            call symput('_blankn', compress(put(len,best.)));

        end;

        drop _firtitl _secndtitl _upcas len;

run;

```


ods listing close;

* most set up in template others below;

* title arial 12pt bold with 12pt paragraph space below;

* all headers to be arial 11pt bold;

* data arial 10pt;

* headers to be central, text values left aligned and numeric centered around decimal point;

/* Update with your variables as needed */

proc report data = comp headline headskip missing nowd split = '\$';;

column flag page paramn paramc ord avisit statord stat

%if &par= 0 %then %do;

("THSm2.2 \$(N=&trt1) &linebot" or1) ("mCC \$(N=&trt2) &linebot" or2) ("SA \$(N=&trt3) &linebot" or3)
/* 7) JMH 07Oct2014 */

(

"Overall Safety\$(N=&trt99) &linebot" or99);;

%end;

%else %do;

("THSm2.2 \$(N=&trt1) &linebot" or1 c1) ("mCC \$(N=&trt2) &linebot" or2 c2) ("SA \$(N=&trt3) &linebot"
or3 c3) /* 7) JMH 07Oct2014 */

(

"Overall Safety\$(N=&trt99) &linebot" or99 c99);;

%end;

define flag / order order = internal noprint;

define page / order order = internal noprint;

define paramn / order order = internal noprint;

define ord / order order = internal noprint;

```

        define statord    / order order = internal noprint;

        define paramc    / group style={just=left cellwidth=0.3 cm} style(header)={just=left}
'Parameter$(units)';

        define avisit    / group style={just=left cellwidth=0.5 cm} style(header)={just=left}
'Study$Day';

        define stat      / display style={just=left cellwidth=0.3 cm} style(header)={just=left}
'Statistic';

%if &par= 0 %then %do;

/*          %if &paramc.='Interpretation' %then %do;*/

        define or1      / display style={just=left cellwidth=0.3 cm}
style(header)={just=left} 'Raw value';

        define or2      / display style={just=left cellwidth=0.3 cm}
style(header)={just=left} 'Raw value';

        define or3      / display style={just=left cellwidth=0.3 cm}
style(header)={just=left} 'Raw value';

        define or99     / display style={just=left cellwidth=0.3 cm}
style(header)={just=left} 'Raw value';

        %end;

        %else %do;

        define or1      / display style={just=left cellwidth=0.25 cm}
style(header)={just=left} 'Raw$value';

        define c1       / display style={just=left cellwidth=0.25 cm}
style(header)={just=left};

        define or2      / display style={just=left cellwidth=0.25 cm}
style(header)={just=left} 'Raw$value';

        define c2       / display style={just=left cellwidth=0.25 cm}
style(header)={just=left};

        define or3      / display style={just=left cellwidth=0.25 cm}
style(header)={just=left} 'Raw$value';

        define c3       / display style={just=left cellwidth=0.25 cm}
style(header)={just=left};

```

```
define or99 / display style={just=left cellwidth=0.25 cm}  
style(header)={just=left} 'Raw$value';
```

```
define c99 / display style={just=left cellwidth=0.25 cm}  
style(header)={just=left};
```

```
%end;
```

```
break before flag / page %if &i=1 %then %do;
```

```
contents="&_fsrtitl" %end; %else %do; contents=" %end;;
```

```
break after page / page;
```

```
compute after ord;
```

```
line " ";
```

```
endcomp;
```

```
compute before page / style={protectspecialchars=off};;
```

```
line "&linetop";
```

```
endcomp;
```

```
compute before _page_ / style={just=left protectspecialchars=off};
```

```
line "\b\fs24\sa24&_FSRTITL." ; * \b = bold, \fs24 is font size 12pt, \sa24 is space after 12pt;
```

```
line "&linebot";
```

```
line "\b\fs24\sa24&_SECTITL." ; * \b = bold, \fs24 is font size 12pt, \sa24 is space after 12pt;
```

```
endcomp;
```

```
compute after _page_ / style={just=left protectspecialchars=off pretext="&linetop."};
```

```
line 'Note: mCC = Menthol conventional cigarettes; SA = Smoking abstinence; THSm2.2  
= Tobacco Heating System 2.2 Menthol.';
```

```
line 'Note: NCS = not clinically significant; CS = clinically significant.';
```

```
LINE 'Note: Change is change from baseline, where baseline is defined as the last  
assessment prior to first randomized product use in mCC / THS 2.2 Menthol arms'; /* 5) JMH  
07Oct2014 */
```

```
line 'or the last assessment prior to 10 AM on Day 1 in the SA arm.';
```

```
line ' ';
```

```
line 'Appendix 15.3.6.10';
```

```
line "Study ID:ZRHM-REXA-08-US Program:&TFLprg Status: &status" &_blankn.*"\~\~"  
"&sysdate" &_blankn.*"\~\~" "(Page &i of &page)";
```

```
endcomp;
```

```
run;
```

```
%end;
```

```
ods rtf close;
```

```
ods results on;
```

```
ods path sashelp.tmplmst (read);
```

```
%mend ;
```

```
%outrtf(blankn=40, halfblk=N);
```

```
ods listing;
```

```
ods listing close;
```

```
proc datasets lib=work nolist memtype=data kill; quit;
```

```
%m_logchk;
```

*=====;

* END OF PROGRAM CODE ;

*=====;