

```

/*=====
| Covance Study Number      : 000000106343      |
| Program Name              : t_labchem.sas      |
| Purpose                   : Summary of Clinical Chemistry Parameters - Safety Population |
| Input Data                : ADSL, ADLB         |
|
| Output Data               : T_15_02_06_16      |
|
| Macros Called             : m_printto, m_logchk |
| Originally Performed by : kpothuri            |
| Date                     : 20MAY2015          |
|
|=====
| Modification History
|-----
| Modified by              :
| Modification Date       :
| Modification Description :
|=====*/
options notes source source2 nfullstimer validvarname=upcase missing=' ' NOQUOTELNMAX spool replace;
ods _all_ close;
ods listing;

%m_printto(route=YES);

*=====;
* START OF PROGRAM CODE                               ;
*=====;

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

/* 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",""));
    call symput('TFLprg',reverse(scan(strip(reverse(compress("&_SASPROGRAMFILE",""))),1,"/"))));
run;

%let tflno=T_15_02_06_16;
%let title1 = Table 15.2.6.16 Summary of Clinical Chemistry Parameters - Safety Population;

/*Chemistry data*/
*reference ranges - post-rand;
data lb_ref;
    set adam.adlb;
    where safaf1 = 'Y' and parcat1='Clinical Chemistry' and anl01f1='Y' and DTYPE^="LOCF" and
paramcd in ("ALT","AST","BILI","BILDIR","ALP","GGT","LDL","GLUC","CREAT","PROT","BUN","TRIG","CHOL","ALB","SODIUM","K");
    if paramcd="GLUC" and LBFAST in ("","N") then delete;

    if avisitn le 99 then delete; *should not have screening;
    if ABLFL="Y" then do;
        avisit="Baseline";
        avisitn=10;
    end;
    if avisit ="Day 0" then delete;

    if not index(param,'(') then param=upcase(substr(param,1,1))||lowcase(substr(param,2));
    else if index(param,'(') then param=upcase(substr(param,1,1))||lowcase(scan(substr(param,2),1,'('))||'('||scan(param,2,'(');
run;

proc freq data=lb_ref noprint;
    tables paramn*param*anrlo*anrhi*sexc /out=range_f (drop=count percent);
run;
proc sort data=range_f out=range_f1(rename=sexc=sexc1) dupout=dup(rename=(sexc=sexc2)) nodupkey; by paramn paramn anrlo anrhi; run;
data range;
    merge range_f1(in=a) dup(in=b);
    by paramn param anrlo anrhi;

    if a and b then flag="Y";
run;

```

```

proc freq data=range noprint;
  table param/out=param_count (where=(count=1)) nocum nopercnt;
run; *for labeling parameters with the reference ranges where the count is one - may need to check;
proc sort data=range; by param; run;
data range_count_post;
  merge param_count(in=a) range(in=b);
  by param;
  if a;
run;

*reference ranges - pre-rand;
data lb_ref_pre;
  set adam.adlb;
  where safbfl = 'Y' and parcat1='Clinical Chemistry' and anl01f1='Y' and DTYPE^="LOCF" and
  paramcd in ('ALT',"AST","BILI","BILDIR","ALP","GGT","LDL","GLUC","CREAT","PROT","BUN","TRIG","CHOL","ALB","SODIUM","K");
  if paramcd="GLUC" and LBFAST in ("","N") then delete;

  if avisitn <=100; *only screening and Day 0;
run;

proc freq data=lb_ref_pre noprint;
  tables paramn*param*anrlo*anrhi*sexc /out=range_f_pre (drop=count percent);
run;
proc sort data=range_f_pre out=range_f1_pre(rename=sexc=sexc1) dupout=dup(rename=(sexc=sexc2)) nodupkey; by paramn paramn anrlo anrhi; run;
data range_pre;
  merge range_f1_pre(in=a) dup(in=b);
  by paramn param anrlo anrhi;

  if a and b then flag="Y";
run;

proc freq data=range_pre noprint;
  table param/out=param_count_pre (where=(count=1)) nocum nopercnt;
run; *for labeling parameters with the reference ranges where the count is one - may need to check;
proc sort data=range_pre; by param; run;
data range_count_pre;
  merge param_count_pre(in=a) range_pre(in=b);
  by param;
  if a;
run;

/*Use ADSL to get N numbers for column headers - pre-rand and post-rand*/
%macro R (saf=, tot=, lb=, where=, outds=, spd=);
data adsl;
  set adam.adsl;
  where &saf = 'Y';
  output;
  trt01an=99;
  trt01a='Overall Safety';
  output;
run;

proc freq data=adsl noprint;
  table trt01an*trt01a/ out =tot(drop=percent rename=(trt01an=trtan trt01a=trta));
run;

data dumtrts; /*Use this to output any columns for which N=0*/ *not needed and did not do for pre-rand;
  attrib trta length =$40.
  trtan length=8.;
  trtan=4;
  trta='THSm2.2';
  output;
  trtan=5;
  trta='mCC';
  output;
  trtan=3;
  trta='SA';
  output;
run;

%global trt_pre3 trt_pre4 trt_pre5 trt_pre96 trt_pre99 trt_post3 trt_post4 trt_post5 trt_post99;
proc sort data=dumtrts; by trtan trta; run;
data &tot;
  merge tot(in=a) dumtrts(in=b); *to output any columns for which N=0;
  by trtan trta;

```

```

if a or b;
if b and not a then count=0;
call symput("trt&spd" || compress(put(trtan,best.)), compress(put(count,best.)));
    rename count=total;
run;

data &lb;
set adam.adlb(where=(parcat1='Clinical Chemistry' and anl01fl='Y' and DTYPE^="LOCF" and
paramcd in ("ALT","AST","BILI","BILDIR","ALP","GGT","LDL","GLUC","CREAT","PROT","BUN","TRIG","CHOL","ALB","SODIUM","K")
and &where));

if paramcd="GLUC" and LBFAST in ("","N") then delete;
if missing(trta) then delete;

%if &saf=safbfl %then %do;
    if avisitn <=100; *only screening and Day 0;
%end;
%if &saf=safaf1 %then %do;
    if avisitn le 99 then delete; *should not have screening;
    if ABLFL="Y" then do;
        avisit="Baseline";
        avisitn=10;
    end;
    if avisit ="Day 0" then delete;
%end;

if not index(param,'(') then param=upcase(substr(param,1,1))||lowcase(substr(param,2));
else if index(param,'(') then param=upcase(substr(param,1,1))||lowcase(scan(substr(param,2),1,'('))||'('||scan(param,2,'(');
output;

trtan=99;
trta='Overall Safety';
output;
run;

*BLOQ, ALOQ;
data n;
length stat $100.;
set &lb;
if index(avalc, "<") > 0 then do;
    statord = 1.3;
    stat = "BLOQ, n (%)";
output;
end;
if index(avalc, ">") > 0 then do;
    statord = 1.6;
    stat = "ALOQ, n (%)";
output;
end;
run;

proc sort data=n out=n1 nodupkey dupout=dup;
    by usubjid trta trtan param paramn avisitn statord;
run;
proc sort data=n1 out=n2;
    by trtan trta paramn param paramcd avisitn avisit statord stat;
run;
proc univariate data=n2 noprint;
    var statord;
    by trtan trta paramn param paramcd avisitn avisit statord stat;
    output out=q_ab&spd N=count; *****KP 8/5;
run;
proc sort data=q_ab&spd; by trtan trta; run;
data OQ;
    merge q_ab&spd &tot; *adsl counts;
    by trtan trta;

    if statord=1.6; *****KP 8/5;
run;
data q_ab_set;
length statval $20.;
set OQ;
where paramn ne .;

if count=0 then statval = strip(put(count,best.)) ;

if count lt 10 then count1=' ' || compress(put(count,best.));
else count1=strip(put(count,best.));

```

```

count1=trim(count1);

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

if count=0 then do;
    statval = ' 0          ' ;
end;
else do;
    if percent=100 then statval = strip(put(count,best.)) || ' (100)' ;
    else if percent lt 10 then statval = strip(count1) || ' (' || strip(put(round(percent,0.1),5.1)) || ')';
    else if percent ge 10 then statval = strip(count1) || ' (' || strip(put(round(percent,0.1),5.1)) || ')';
end;

    drop count count1 total percent;
run;
proc sort data=q_ab_set; by paramn param avisitn avisit stat statord; run;
proc transpose data=q_ab_set out=&outds(drop=_) prefix=t;
    by paramn param avisitn avisit stat statord;
    var statval;
    id trtan;
    idlabel trta;
run;
data &outds;
    set &outds;
    where paramn ne .;
run;
%mend R;
%R (saf=safb1, tot=pre_tot, lb=lb_pre, where=%str(safb1="Y"), outds=0Q_pre_aloq, spd=_pre); *pre-rand; ***kp;
%R (saf=safa1, tot=post_tot, lb=lb_post, where=%str(safa1 = 'Y'), outds=0Q_post_aloq, spd=_post); *post-rand; ***kp;

*statistics - pre-rand;
data adlb_orig_pre;
    set lb_pre;
    statval=aval;
run;
proc sort data=adlb_orig_pre;
    by trtan trta paramn param paramcd avisitn avisit;
run;

*statistics - Change from baseline to repeat after every analysis day - post-rand;
data adlb_orig adlb_chg1 adlb_chg2 adlb_chg3 adlb_chg4 adlb_chg5;
    set lb_post;
    statval=aval;
    output adlb_orig;
    if avisitn = 106 then do;
        avisit='Change from Baseline at Day 6';
        avisitn=107;
        statval=chg;
        output adlb_chg1;
    end;
    if avisitn = 130 then do;
        avisit='Change from Baseline at Day 30';
        avisitn=131;
        statval=chg;
        output adlb_chg2;
    end;
    if avisitn = 160 then do;
        avisit='Change from Baseline at Day 60';
        avisitn=161;
        statval=chg;
        output adlb_chg3;
    end;
    if avisitn = 190 then do;
        avisit='Change from Baseline at Day 90';
        avisitn=190.1;
        statval=chg;
        output adlb_chg4;
    end;
    if avisitn = 191 then do;
        avisit='Change from Baseline at Day 91';
        avisitn=191.1;
        statval=chg;
        output adlb_chg5;
    end;
end;

```

```

run;

data adlb_all;
  set adlb_orig adlb_chg1 adlb_chg2 adlb_chg3 adlb_chg4 adlb_chg5;
run;

proc sort data=adlb_all;
  by trtan trta;
run;

data all;
  merge adlb_all(in=a) dumtrts(in=b); *to output any columns for which N=0;
  by trtan trta;
  if a or b;
  if b and not a then statval=.;
run;

proc sort data=all;
  by trtan trta paramn param paramcd avisitn avisit;
run;

*summary statistics for pre-rand and post-rand;
%macro stat (dsin=, dsout=, oqin=, outds=);
proc univariate data=&dsin noprint;
  var statval;
  by trtan trta paramn param paramcd avisitn avisit;
  output out=results01 N=N1 mean=mean1 std=std1 median=med1 min=min1 max=max1;
run;

data results02;
  set results01;
  attrib meansd minmax n median length=$20.;

  *decimals depending on the type of parameter;
  if paramn in (4,9) then do; *direct bilirubin, creatinine;
    n = left(compress(put(n1,8.)));
    if not missing(med1) then median = left(compress(put(round(med1,0.001),8.3)));
    if not missing(mean1) and not missing(std1) then meansd = left(compress(put(round(mean1,0.001),8.3))) || ' (' || compress(p
ut(0.0001*ceil(std1*10000),8.4)) || ')';
    else if missing(std1) and not missing(mean1) then meansd = left(compress(put(round(mean1,0.001),8.3)));
    if not missing(min1) and not missing(max1) then minmax = left(compress(put(round(min1,0.01),8.2))) || ', ' || left(compress
(put(round(max1,0.01),8.2)));
  end;
  else if paramn in (3,10,16) then do; *Bilirubin, protein, potassium;
    n = left(compress(put(n1,8.)));
    if not missing(med1) then median = left(compress(put(round(med1,0.01),8.2)));
    if not missing(mean1) and not missing(std1) then meansd = left(compress(put(round(mean1,0.01),8.2))) || ' (' || compress(pu
t(0.001*ceil(std1*1000),8.3)) || ')';
    else if missing(std1) and not missing(mean1) then meansd = left(compress(put(round(mean1,0.01),8.2)));
    if not missing(min1) and not missing(max1) then minmax = left(compress(put(round(min1,0.1),8.1))) || ', ' || left(compress(
put(round(max1,0.1),8.1)));
  end;
  else do;
    n = left(compress(put(n1,8.)));
    if not missing(med1) then median = left(compress(put(round(med1,0.1),8.1)));
    if not missing(mean1) and not missing(std1) then meansd = left(compress(put(round(mean1,0.1),8.1))) || ' (' || compress(put
(0.01*ceil(std1/0.01),8.2)) || ')';
    else if missing(std1) and not missing(mean1) then meansd = left(compress(put(round(mean1,0.1),8.1)));
    if not missing(min1) and not missing(max1) then minmax = left(compress(put(round(min1,1.),8.))) || ', ' || left(compress(pu
t(round(max1,1.),8.)));
  end;

  ***kp 8/5;
  if n1<4 then do;
    meansd="NC";
    minmax="NC";
    median="NC";
  end;

  drop n1 mean1 std1 med1 min1 max1;
run;

proc sort data=results02;
  by paramn param avisitn avisit;
run;
proc transpose data=results02 out=results04 prefix=t name=varname;
  by paramn param avisitn avisit;

```

```

var n meansd median minmax;
id trtan;
idlabel trta;
run;

data &dsout;
set results04;
by paramn param avisitn avisit;
attrib stat length = $100.;

if first.avisitn then statord=1;
else statord+1;
if varname='N' then stat='n';
else if varname='MEANSD' then stat='Mean (SD)';
else if varname='MEDIAN' then stat='Median';
else if varname='MINMAX' then stat='Min, Max';

if paramn=. then delete;
drop varname;
run;

*****kp 8/6;
proc sort data=&oqin; by trtan paramn avisitn; run;
proc sort data=results01; by trtan paramn avisitn; run;
data oq1;
merge &oqin(in=a where=(statord=1.3)) results01(keep=trtan n1 paramn avisitn rename=(n1=total));
by trtan paramn avisitn;
if a;
run;

data oq2;
length statval $20.;
set oq1;
if count=0 then statval = strip(put(count,best.)) ;

if count lt 10 then count1=' ' || compress(put(count,best.));
else count1=strip(put(count,best.));

count1=trim(count1);

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

if count=0 then do;
statval = ' 0 ' ;
end;
else do;
if percent=100 then statval = strip(put(count,best.)) || ' (100)' ;
else if percent lt 10 then statval = strip(count1) || ' (' || strip(put(round(percent,0.1),5.1)) || ')';
else if percent ge 10 then statval = strip(count1) || ' (' || strip(put(round(percent,0.1),5.1)) || ')';
end;

drop count count1 total percent;
run;

proc sort data=oq2; by paramn param avisitn avisit stat statord; run;
proc transpose data=oq2 out=&outds(drop=_) prefix=t;
by paramn param avisitn avisit stat statord;
var statval;
id trtan;
idlabel trta;
run;

data &outds;
set &outds;
where paramn ne .;
run;

%mend stat;
%stat (dsin=adlb_orig_pre, dsout=stat_pre, oqin=q_ab_pre, outds=oq_pre_bloq); *pre-rand; ***kp;
%stat (dsin=all, dsout=stat_post, oqin=q_ab_post, outds=oq_post_bloq); *post-rand; ***kp;

/*Obtaining categorical stats - without change in baseline repetition*/
*pre-rand;
data adlb_orig2_pre;

```

```

    set lb_pre;
    statval=anrind;
run;
proc sort data=adlb_orig2_pre; by trtan trta paramn param paramcd avisitn avisit aclsig; run;

*post-rand;
data adlb_orig2;
    set lb_post;
    statval=anrind;
run;

proc sort data=adlb_orig2; by trtan trta; run;
data all2;
    merge adlb_orig2(in=a) dumtrts(in=b); *to output any columns for which N=0;
    by trtan trta;
    if a or b;
    if b and not a then statval='';
run;
proc sort data=all2; by trtan trta paramn param paramcd avisitn avisit aclsig; run;

*categorical stats for pre-rand and post-rand;
%macro cat (tot=, dsin=, OQ=, stat=, dsout=);
proc freq data=&dsin noprint;
    by trtan trta paramn param paramcd avisitn avisit;
    tables statval / out=stats(drop=percent);
run;

data stats (drop=STATVAL rename=(STATVAL_=STATVAL));
    set stats;
    length STATVAL_ $50.;
    STATVAL_=statval;
run;

data all2_test;
    set &dsin;
    if aclsig in ('CS','NCS');
    if aclsig='CS' then statvaldum='Y';
    if aclsig='NCS' then statvaldum='A';
run;

proc freq data=all2_test noprint;
    by trtan trta paramn param paramcd avisitn avisit aclsig;
    tables statvaldum / out=statsab(drop=percent);
run;

data statsab2;
    set statsab;
    attrib statval length=$50.;
    if aclsig='NCS' then statval='ABNORMAL NCS';
    if aclsig='CS' then statval='ABNORMAL CS';
    drop aclsig;
run;

data stats2a;
    set stats statsab2;
run;

proc sort data=stats2a; by trtan trta; run;
data stats2;
    merge stats2a &tot; *adsl counts;
    by trtan trta;
run;

data stats3;
    set stats2(rename=(statval=statistic));
    format statval $20. stat $100.;

    if statistic='LOW' then do;
        stat='Low value - n (%)';
        statord=5;
    end;
    else if statistic='NORMAL' then do;
        stat='Normal value - n (%)';
        statord=6;
    end;
    else if statistic='HIGH' then do;
        stat='High value - n (%)';

```

```

    statord=7;
end;
else if statistic='ABNORMAL CS' then do;
    stat='Abnormal CS - n (%)';
    statord=8;
end;
else if statistic='ABNORMAL NCS' then do;
    stat='Abnormal NCS - n (%)';
    statord=8.05;
end;

if count=0 then statval = strip(put(count,best.)) ;

if count lt 10 then count1=' ' || compress(put(count,best.));
else count1=strip(put(count,best.));

count1=trim(count1);

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

if count=0 then do;
    statval = ' 0 ' ;
end;
else do;
    if percent=100 then statval = strip(put(count,best.)) || ' (100) ' ;
    else if percent lt 10 then statval = strip(count1) || ' (' || strip(put(round(percent,0.1),5.1)) || ')';
    else if percent ge 10 then statval = strip(count1) || ' (' || strip(put(round(percent,0.1),5.1)) || ')';
end;

    drop count count1 total percent;
run;

proc sort data=stats3 out=stats3a(where=(not missing(statistic) and not missing(param)));
    by paramn param avisitn avisit stat statord;
run;
proc transpose data=stats3a out=stats4(drop=_) prefix=t;
    by paramn param avisitn avisit stat statord;
    var statval;
    id trtan;
    idlabel trta;
run;

*dummy records for category - low/min/high value and abnormal CS;
proc sort data=stats3(where=(not missing(param))) out=stats3b(keep=paramn param avisit avisitn) nodupkey;
    by paramn param avisit avisitn;
run;
data extra (drop=j);
    set stats3b;
    format stat avisit $40.;
    by paramn param avisit avisitn;

%if &dsin=adlb_orig2_pre %then %do;
    do j=5 to 8;
        if j=5 then stat='Low value - n (%)';
        else if j=6 then stat='Normal value - n (%)';
        else if j=7 then stat='High value - n (%)';
        else if j=8 then stat='Abnormal CS - n (%)';
        statord=j;
        output;
    end;
%end;
%if &dsin=all2 %then %do;
    do j=5 to 8;
        if j=5 then stat='Low value - n (%)';
        else if j=6 then stat='Normal value - n (%)';
        else if j=7 then stat='High value - n (%)';
        else if j=8 then stat='Abnormal CS - n (%)';
        statord=j;
        output;
    end;
%end;
run;
data extra_1;
    set stats3b;
    by paramn param avisit avisitn;
    format stat $40.;

```



```

statord=8.05;
stat='Abnormal NCS - n (%)';
run;

proc sort data=stats4;
    by paramn param avisit avisitn statord stat;
run;
proc sort data=extra;
    by paramn param avisit avisitn statord stat;
run;
proc sort data=extra_1;
    by paramn param avisit avisitn statord stat;
run;
data stats5; *end of categories stats;
    merge stats4 extra extra_1;
    by paramn param avisit avisitn statord stat;
run;

*flag for missing counts - n;
data part1;
    set &OQ &stat; *adding on OQ if present & adding on stats;
run;
proc sort data=part1; by paramn param avisit avisitn statord; run;

proc sort data=part1(where=(stat='n')) out=missflags;
    by paramn param avisit avisitn;
run;
data missflags2;
    set missflags;

    if strip(t3)='' then mflag3=1;
    if strip(t4)='' then mflag4=1;
    if strip(t5)='' then mflag5=1;
    %if &dsin=adlb_orig2_pre %then %do; if strip(t96)='' then mflag96=1; %end;
    if strip(t99)='' then mflag99=1;

    drop t: stat:;
run;

proc sort data=part1; by paramn param avisit avisitn; run;
proc sort data=missflags2; by paramn param avisit avisitn; run;
proc sort data=stats5; by paramn param avisit avisitn; run;
data results05_a;
    merge part1 missflags2;
    by paramn param avisit avisitn;
run;

data stats5a;
    merge stats5 missflags2(where=(avisitn not in(107,131,161,190.1,191.1))); *change in baseline;
    by paramn param avisit avisitn;
    if stat='n' then delete;
run;

data &dsout;
    set results05_A stats5A;

    %if &stat=stat_pre %then %do;
    array a[5] t3 t4 t5 t_96 t99;
    array b[5] mflag3 mflag4 mflag5 mflag96 mflag99;
    do i=1 to 5;
        if statord>4 and b[i] ne 1 then if missing(a[i]) then a[i]='0';
    end;
    if stat='n' and statord=1 then do;
        if missing(t3) then t3='0';
        if missing(t4) then t4='0';
        if missing(t5) then t5='0';
        if missing(t96) then t96='0';
        if missing(t99) then t99='0';
    end;
    %end;
    %if &stat=stat_post %then %do;
    array a[4] t3 t4 t5 t99;
    array b[4] mflag3 mflag4 mflag5 mflag99;
    do i=1 to 4;
        if statord>4 and b[i] ne 1 then if missing(a[i]) then a[i]='0';
    end;
    if stat='n' and statord=1 then do;

```

```

    if missing(t3) then t3='0';
    if missing(t4) then t4='0';
    if missing(t5) then t5='0';
    if missing(t99) then t99='0';
end;
%end;
run;
proc sort data=&dsout; *combine OQ, stats, categories;
    by paramn param avisitn avisit statord;
run;
%mend cat;
%cat (tot=pre_tot, dsin=adlb_orig2_pre, OQ=%str(OQ_pre_bloq OQ_pre_aloq), stat=stat_pre, dsout=pre_rand); *pre-rand; ***kp;
%cat (tot=post_tot, dsin=all2, OQ=%str(OQ_post_bloq OQ_post_aloq), stat=stat_post, dsout=post_rand); *post-rand; ***kp;

*Shift in normality;
proc sort data=lb_post out=adlb_norm(rename=(shift1=statistic)); where shift1 ne ""; by trtan trta paramn param paramcd avisitn avis
it; run;
proc freq data=adlb_norm noprint;
    by trtan trta paramn param paramcd avisitn avisit;
    tables statistic / out=s_norm(drop=percent);
run;
proc sort data=s_norm; by trtan trta; run;
data s_norm_1;
    merge s_norm post_tot; *adsl counts;
    by trtan trta;
run;
data s_norm_2;
    set s_norm_1;
    format statistic $50. stat $100.;

if statistic='Low to Low' then do;
    stat='Low to Low - n (%)';
    statord=9;
end;
else if statistic='Low, NCS to Low, NCS' then do;
    stat='Low, NCS to Low, NCS - n (%)';
    statord=10;
end;
else if statistic='Low, NCS to Normal' then do;
    stat='Low, NCS to Normal - n (%)';
    statord=11;
end;
else if statistic='Low to Normal' then do;
    stat='Low to Normal - n (%)';
    statord=12;
end;
else if statistic='Low, NCS to High, NCS' then do;
    stat='Low, NCS to High, NCS - n (%)';
    statord=13;
end;
else if statistic='Normal to Low' then do;
    stat='Normal to Low - n (%)';
    statord=14;
end;
else if statistic='Normal to Low, NCS' then do;
    stat='Normal to Low, NCS - n (%)';
    statord=15;
end;
else if statistic='Normal to Low, CS' then do;
    stat='Normal to Low, CS - n (%)';
    statord=16;
end;
else if statistic='Normal to Normal' then do;
    stat='Normal to Normal - n (%)';
    statord=17;
end;
else if statistic='Normal to High' then do;
    stat='Normal to High - n (%)';
    statord=18;
end;
else if statistic='Normal to High, NCS' then do;
    stat='Normal to High, NCS - n (%)';
    statord=19;
end;
else if statistic='Normal to High, CS' then do;
    stat='Normal to High, CS - n (%)';
    statord=20;
end;

```

```

end;
else if statistic='High to Low' then do;
  stat='High to Low - n (%)';
  statord=21;
end;
else if statistic='High, NCS to Normal' then do;
  stat='High, NCS to Normal - n (%)';
  statord=22;
end;
else if statistic='High to Normal' then do;
  stat='High to Normal - n (%)';
  statord=23;
end;
else if statistic='High, NCS to High, NCS' then do;
  stat='High, NCS to High, NCS - n (%)';
  statord=24;
end;
else if statistic='High, NCS to High, CS' then do;
  stat='High, NCS to High, CS - n (%)';
  statord=25;
end;
else if statistic='High to High' then do;
  stat='High to High - n (%)';
  statord=26;
end;
else if statistic='High, CS to High, NCS' then do;
  stat='High, CS to High, NCS - n (%)';
  statord=27;
end;
if count=0 then statval = strip(put(count,best.)) ;
if count lt 10 then count1=' ' || compress(put(count,best.));
else count1=strip(put(count,best.));

count1=trim(count1);

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

if count=0 then do;
  statval = ' 0 ' ;
end;
else do;
  if percent=100 then statval = strip(put(count,best.)) || ' (100)' ;
  else if percent lt 10 then statval = strip(count1) || ' (' || strip(put(round(percent,0.1),5.1)) || ' )';
  else if percent ge 10 then statval = strip(count1) || ' (' || strip(put(round(percent,0.1),5.1)) || ' )';
end;

  drop count count1 total percent;
run;
proc sort data=s_norm_2 out=s_norm_3(where=(not missing(statistic) and not missing(param)));
  by paramn param avisit avisitn stat statord;
run;
proc transpose data=s_norm_3 out=s_norm_4(drop=_) prefix=t;
  by paramn param avisit avisitn stat statord;
  var statval;
  id trtan;
  idlabel trta;
run;
proc sort data=s_norm_4;
  by paramn param avisit avisitn stat statord;
run;

```

*Shift in Toxicity and Toxicity Grades;

```

%macro tox (lb=, fix=);
proc sort data=&lb; by trtan trta paramn param paramcd avisitn avisit; run;
data tox1 tox2 tox3;
set &lb;
if paramcd in ("ALP","ALT","AST","BILI","BUN","CHOL","CREAT","GGT","TRIG") and
parcat4n = 1 then output tox1;
if paramcd in ("ALB","PROT") and
parcat4n = 2 then output tox2;
if paramcd in ("GLUC","K","SODIUM") and
parcat4n = 3 then output tox3;
run;

```

*macro for Toxicity Grades;

```

%macro toxg (dset=, b=, pb=, name=, outd=);
data tox_base(rename=(&b=statistic)) tox_pb(rename=(&pb=statistic));
length &b &pb $50.;
set &dset;
rename &name = tox;
%if &fix=_pre %then %do; if avisitn = 100 and ablf1="Y" then output tox_base; %end;
%if &fix=_post %then %do; if avisitn = 10 then output tox_base; %end;

%if &fix=_pre %then %do; else if avisitn ne 100 or (avisitn = 100 and ablf1=" ") then output tox_pb; %end;
%if &fix=_post %then %do; else if avisitn ne 10 then output tox_pb; %end;
run;
proc freq data=tox_base noprint;
by trtan trta paramn param paramcd avisitn avisit tox;
tables statistic / out=tox_base_1(drop=percent);
run;
proc freq data=tox_pb noprint;
by trtan trta paramn param paramcd avisitn avisit tox;
tables statistic / out=tox_pb_1(drop=percent);
run;
data &outd;
set tox_base_1(in=a) tox_pb_1(in=b);
run;
data &outd&fix;
set &outd;
toxord=1;
run;
%mend toxg;
%toxg (dset=tox1, b=BTOXGRH, pb=ATOXGRH, name=LBTOXH, outd=high);
%toxg (dset=tox2, b=BTOXGRL, pb=ATOXGRL, name=LBTOXL, outd=low);
%toxg (dset=tox3, b=BTOXGRH, pb=ATOXGRH, name=LBTOXH, outd=bi_high);
%toxg (dset=tox3, b=BTOXGRL, pb=ATOXGRL, name=LBTOXL, outd=bi_low);

*macro for Shift in Toxicity - post-rand only;
%if &lb=lb_post %then %do;
%macro toxsh (dset=, sh=, name=, outd=);
data tox_sh(rename=(&sh=statistic));
length &sh $50.;
set &dset;
rename &name = tox;
run;
proc freq data=tox_sh noprint;
where statistic ne "";
by trtan trta paramn param paramcd avisitn avisit tox;
tables statistic / out=&outd(drop=percent);
run;
data &outd&fix;
set &outd;
toxord=2;
run;
%mend toxsh;
%toxsh (dset=tox1, sh=SHIFT2, name=LBTOXH, outd=high_sh);
%toxsh (dset=tox2, sh=SHIFT3, name=LBTOXL, outd=low_sh);
%toxsh (dset=tox3, sh=SHIFT2, name=LBTOXH, outd=bi_high_sh);
%toxsh (dset=tox3, sh=SHIFT3, name=LBTOXL, outd=bi_low_sh);
%end;

%mend tox;
%tox (lb=lb_pre, fix=_pre); *pre-rand;
%tox (lb=lb_post, fix=_post); *post-rand;

*Toxicity Grades for pre-rand;
data pre_tox;
set high_pre low_pre bi_high_pre bi_low_pre;
run;

proc sort data=pre_tox;
by trtan trta paramn param paramcd avisitn avisit tox statistic toxord;
run;
proc univariate data=pre_tox noprint;
var count;
by trtan trta paramn param paramcd avisitn avisit tox statistic toxord;
output out=pre_tox_c sum=count;
run;

*Shift in Toxicity and Toxicity Grades for post-rand;

```

```

data post_tox;
  set high_post low_post bi_high_post bi_low_post high_sh_post low_sh_post bi_high_sh_post bi_low_sh_post;
run;

%macro tox_all (inds=, tot=, outds=);
proc sort data=&inds; by trtan trta; run;
data tox_all1;
  merge &inds &tot; *adsl counts;
  by trtan trta;
run;
data tox_all2;
  set tox_all1;
  format statistic $50. stat $100.;

  if toxord = 1 then do;
    if statistic='0' then do;
      stat='   Grade 0 - n (%)';
      statord=50;
    end;
  else if statistic='1' then do;
    stat='   Grade 1 - n (%)';
    statord=51;
  end;
  else if statistic='2' then do;
    stat='   Grade 2 - n (%)';
    statord=52;
  end;
  else if statistic='3' then do;
    stat='   Grade 3 - n (%)';
    statord=53;
  end;
  end;
  if toxord = 2 then do;
    stat="      || strip(statistic) || " - n (%)";
    statord=100;
  end;

  if count=0 then statval = strip(put(count,best.)) ;

  if count lt 10 then count1=' ' || compress(put(count,best.));
  else count1=strip(put(count,best.));

  count1=trim(count1);

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

  if count=0 then do;
    statval = ' 0          ' ;
  end;
  else do;
    if percent=100 then statval = strip(put(count,best.)) || ' (100)' ;
    else if percent lt 10 then statval = strip(count1) || ' (' || strip(put(round(percent,0.1),5.1)) || ')';
    else if percent ge 10 then statval = strip(count1) || ' (' || strip(put(round(percent,0.1),5.1)) || ')';
  end;

  drop count count1 total percent;
run;
proc sort data=tox_all2 out=tox_all3(where=(not missing(statistic) and not missing(param)));
  by paramn param avisit avisitn tox toxord stat statord;
run;
proc transpose data=tox_all3 out=tox_all4(drop=_) prefix=t;
  by paramn param avisit avisitn tox toxord stat statord;
  var statval;
  id trtan;
  idlabel trta;
run;
proc sort data=tox_all4 out=dum_tox (keep=paramn param avisitn avisit tox toxord) nodupkey;
  by paramn param avisitn avisit tox toxord;
run;
data dum_tox;
length stat $100.;
set dum_tox;
by paramn param avisitn avisit tox toxord;

```

```

stat=tox;
if toxord=1 then do;
  if first.toxord then statord=8.1;
  else statord+0.1;
end;
else if toxord=2 then do;
  if first.toxord then statord=28;
  else statord+0.1;
end;

drop tox;
run;
proc sort data=tox_all4; by paramn param avisitn avisit toxord stat statord; run;
proc sort data=dum_tox; by paramn param avisitn avisit toxord stat statord; run;
data tox_all5;
  merge tox_all4 dum_tox;
  by paramn param avisitn avisit toxord stat statord;

  if tox="" then tox=stat;
run;
proc sort data=tox_all5 out=&outds; by paramn param avisitn avisit toxord tox statord stat; run;
%mend tox_all;
%tox_all (inds=pre_tox_c, tot=pre_tot, outds=pre_tox_1); *pre-rand;
%tox_all (inds=post_tox, tot=post_tot, outds=post_tox_1); *post-rand;

*combine all datasets - pre-rand & post-rand;
data rand_1 (drop=avisit rename=(avisit1=avisit));
length avisit1 $200;
set pre_rand pre_tox_1;
pd=1;
avisit1=avisit;
run;
proc sort data=rand_1; by pd paramn param avisitn avisit toxord tox statord stat; run;
data rand_2;
set post_rand(in=a) s_norm_4(in=b) post_tox_1(in=c);
pd=2;
if a then post=1;
if b then post=2;
if c then post=3;
run;

*shift in normality dummy - labels;
proc sort data=rand_2 out=norm_dum1 (where=(post=2) keep=post avisit avisitn) nodupkey; by post avisitn avisit; run;
data norm_dum2;
length avisit1 $200.;
set norm_dum1;
avisit1 = "Shift in Normality from Baseline to " || strip(avisit);
run;
proc sort data=rand_2; by post avisitn avisit; run;
data rand_2_1;
merge rand_2 norm_dum2;
by post avisitn avisit;
if avisit1 = "" then avisit1 = avisit;

if post=2 then do;
  if avisitn in (106,130,160) then avisitn=avisitn+2;
  if avisitn = 190 then avisitn = 190.2;
  if avisitn = 191 then avisitn = 191.2;
end;
run;

*shift in toxicity dummy - labels;
proc sort data=rand_2_1 out=tox_dum1 (where=(post=3 and toxord=2) keep=post toxord avisit1 avisitn) nodupkey; by post toxord avisitn
avisit1; run;
data tox_dum2;
length avisit2 $200.;
set tox_dum1;
avisit2 = "Shift in Toxicity from Baseline to " || strip(avisit1);
run;
proc sort data=rand_2_1; by post toxord avisitn avisit1; run;
data rand_2_2 (drop=avisit avisit1 rename=(avisit2=avisit));
merge rand_2_1 tox_dum2;
by post toxord avisitn avisit1;
if avisit2 = "" then avisit2 = avisit1;

if post=3 and toxord=2 then do;
  if avisitn in (106,130,160) then avisitn=avisitn+3;

```

```

    if avisitn = 190 then avisitn = 190.3;
    if avisitn = 191 then avisitn = 191.3;
end;
run;
proc sort data=rand_2_2; by pd paramn param avisitn avisit toxord tox statord stat; run;
data rand_2_3;
    set rand_2_2;
    if avisit='Change from Baseline at Day 6' then avisit='Change from Baseline to Day 6/Discharge Confinement';
    if avisit='Change from Baseline at Day 30' then avisit='Change from Baseline to Day 30';
    if avisit='Change from Baseline at Day 60' then avisit='Change from Baseline to Day 60';
    if avisit='Change from Baseline at Day 90' then avisit='Change from Baseline to Day 90';
    if avisit='Change from Baseline at Day 91' then avisit='Change from Baseline to Day 91/Discharge Ambulatory';
run;

data rand_tot;
length paramc_ $200;
set rand_1 rand_2_3;

if statord<=8.05 or 9<=statord<=27 or statord in (50,51,52,52,53,100) then do;
    if t3 = "" then t3 = "0";
    if t4 = "" then t4 = "0";
    if t5 = "" then t5 = "0";
    if pd=1 then do; if t96 = "" then t96 = "0"; end;
    if t99 = "" then t99 = "0";
end;
if pd=1 then do;
    if param="Direct bilirubin (mg/dL)" then param="Direct bilirubin (mg/dL) [0-0.4]";
end;

if pd=2 then do;
    if param="Albumin (g/L)" then param="Albumin (g/L) [33-49]";
    if param="Bilirubin (mg/dL)" then param="Bilirubin (mg/dL) [0.2-1.2]";
    if param="Blood urea nitrogen (mg/dL)" then param="Blood urea nitrogen (mg/dL) [4-24]";
    if param="Direct bilirubin (mg/dL)" then param="Direct bilirubin (mg/dL) [0-0.4]";
    if param="Glucose (mg/dL)" then param="Glucose (mg/dL) [70-100]";
    if param="Potassium (mmol/L)" then param="Potassium (mmol/L) [3.4-5.4]";
end;

paramc_='\b ' || strip(param) || ' \b0';
run;

proc sql noprint;
    create table tflds.&tfldno. as
    select avisit, stat, t4, t5, t3, t96, t99, pd, paramn, avisitn, toxord, tox, statord, param
    from rand_tot
    order by pd, paramn, param, avisitn, avisit, toxord, tox, statord, stat;
quit;

data page;
    set rand_tot;
    by pd paramn param avisitn avisit toxord tox statord stat;

if avisitn in (1,10,100,106,130,160,190,191) and statord<=8.05 then testfl=1;
else if avisitn in (1,10,100,106,130,160,190,191) and statord>8.05 then testfl=2;
else if find(avisit,"Change")>0 then testfl=3;
else if find(avisit,"Shift in Normality")>0 then testfl=4;
else if find(avisit,"Shift in Toxicity")>0 then testfl=5;
run;

proc sort data=page; by pd paramn param avisitn testfl avisit toxord tox statord stat; run;
data final;
set page;
    by pd paramn param avisitn avisit testfl toxord tox statord stat;

if first.testfl then ln=1;
else ln+1;
if ln=1 then page+1;
call symput("page",compress(put(page,best.)));
run;
%put &page;

/* Standard - leave this */
options number nodate orientation=landscape 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=130, halfblank=N);

%if &halfblank=N %then %let halfblank=;
%else %if &halfblank=Y %then %let halfblank=\-;

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 foot
ery=1440 ;
ods noproctitle;
%do i=1 %to &page;

title ;
footnote;
%let wd=0;
ods proclabel = ' ';

data comp;
    set final end=eof;
    where page=&i;

    /* Amend title as needed */
    _firtitl="%title1.";
    _upcas=(length("Path: &TFLpath.")-length(compress("Path:&TFLpath.",'ABCDEFGHIJKLMNOPQRSTUVWXYZ')))/2;
    len=&blankn.-length("(page &i of &page)");
    if eof then do;
        call symput('_FSRTITL', trim(left(_firtitl)));
        call symput('_blankn', compress(put(len,best.)));

        call symput('pd', strip(put(pd,best.)));
        call symput('paramn', strip(put(paramn,best.)));
    end;
    drop _firtitl _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 nowd split = '$' %if &i=1 %then %do; contents=' ' %end; %else %do; contents='' %end;;;
    column page pd paramn param paramc_ avisitn avisit
           toxord tox statord stat
%if &pd=1 %then %do; t4 t5 t3 t96 t99 %end;
%if &pd=2 %then %do; t4 t5 t3 t99 %end;;

define page          /order order = internal noprint;
define pd            /order = internal noprint;
define paramn        /order = internal noprint;
define param         /order = internal noprint;
define paramc_       /"Param (units)$[Reference range]" group style={just=left cellwidth=1.5cm} style(header)={just=left};
define avisitn       /order order = internal noprint;
define avisit        /"Study Day" group style={just=left cellwidth=1.2cm} style(header)={just=left};
define toxord        /order = internal noprint;
define tox           /order = internal noprint;
define statord       /order = internal noprint;
define stat          /"Statistic" display style(column)={asis=on just=left cellwidth=1.9cm} style(header)={just=left};
%if &pd=1 %then %do;
define t4            /"THSm2.2$(N=&trt_pre4)" display style={just=c cellwidth=1.2cm} style(header)={just=center};
define t5            /"mCC$(N=&trt_pre5)" display style={just=c cellwidth=1.2cm} style(header)={just=center};
define t3            /"SA$(N=&trt_pre3)" display style={just=c cellwidth=1.2cm} style(header)={just=center};
    define t96        /"Product Test$(N=&trt_pre96)" display style={just=c cellwidth=1.2cm} style(header)={just=center};
define t99           /"Overall Safety$(N=&trt_pre99)" display style={just=c cellwidth=1.2cm} style(header)={just=center};
%end;
%if &pd=2 %then %do;
define t4            /"THSm2.2$(N=&trt_post4)" display style={just=c cellwidth=1.2cm} style(header)={just=center};
define t5            /"mCC$(N=&trt_post5)" display style={just=c cellwidth=1.2cm} style(header)={just=center};
define t3            /"SA$(N=&trt_post3)" display style={just=c cellwidth=1.2cm} style(header)={just=center};
define t99           /"Overall Safety$(N=&trt_post99)" display style={just=c cellwidth=1.2cm} style(header)={just=center};
%end;

```



```

compute after avisit;
  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 " ";
  %if &pd=1 %then %do; line "Safety Time Period: Pre-Randomized Period"; %end;
  %if &pd=2 %then %do; line "Safety Time Period: Randomized Period"; %end;
  line "&linebot";
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.';
  %if &pd=1 %then %do; line 'Note: Product Test refers to all subjects who tested the THS product on Day -2 but were not randomized.
The Overall Safety refers to all subjects exposed to THSm2.2.'; %end;
  %if &pd ne 1 %then %do; line 'Note: The Overall Safety refers to the safety population post-randomization.'; %end;
  line 'Note: NCS = Not Clinically Significant, CS=Clinically Significant.';
  line 'Note: Percentages are based on the number of subjects indicated in the column header (N).';
  %if &pd=1 %then %do;
    %if &paramn ne 4 %then %do; line 'Note: Normality ranges for this parameter are provided in Appendix.'; %end;
  %end;
  %if &pd=2 %then %do;
    %if &paramn=1 %then %do; line 'Note: Normality ranges for this parameter are provided in Appendix.'; %end;
    %if &paramn=2 %then %do; line 'Note: Normality ranges for this parameter are provided in Appendix.'; %end;
    %if &paramn=5 %then %do; line 'Note: Normality ranges for this parameter are provided in Appendix.'; %end;
    %if &paramn=6 %then %do; line 'Note: Normality ranges for this parameter are provided in Appendix.'; %end;
    %if &paramn=9 %then %do; line 'Note: Normality ranges for this parameter are provided in Appendix.'; %end;
    %if &paramn=10 %then %do; line 'Note: Normality ranges for this parameter are provided in Appendix.'; %end;
    %if &paramn=12 %then %do; line 'Note: Normality ranges for this parameter are provided in Appendix.'; %end;
    %if &paramn=13 %then %do; line 'Note: Normality ranges for this parameter are provided in Appendix.'; %end;
    %if &paramn=15 %then %do; line 'Note: Normality ranges for this parameter are provided in Appendix.'; %end;
    %if &paramn=23 %then %do; line 'Note: Normality ranges for this parameter are provided in Appendix.'; %end;
  %end;
  line '';
  line 'Appendix 15.3.6.6';
  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;

%outtrtf(blankn=36, halfblank=N);

*=====;
* END OF PROGRAM CODE ;
*=====;

%m_logchk;

```