

```

%_mprintto;
options notes nosource;
proc datasets lib=work nolist memtype=data kill; quit;
%put NOTE:
=====;
%put NOTE: Covance Study Number : 000000106324;
%put NOTE: Client Protocol ID   : ZRHR-REXC-03-EU;
%put NOTE: Program Name        : t_laburin.sas;
%put NOTE: Purpose              : Summary of Urinalysis Parameters -
Safety Population;
%put NOTE: ;
%put NOTE: Input Data           : ADAM.ADSL ADAM.ADLB;
%put NOTE: Output               : t_15_02_06_15(urin);
%put NOTE: Macros Called        : _MPRINTTO;
%put NOTE: ;
%put NOTE: Programmed by        : cvn_kbooth;
%put NOTE: Creation Date        : 2014-09-15;
%put NOTE: SAS Version          : 9.3;
%put NOTE: ;
%put NOTE: == Latest Run
=====;
%put NOTE: Run by                : &sysuserid;
%put NOTE: Date/Time             :
%sysfunc(putn(%sysfunc(date()),e8601da.))T%sysfunc(putn(%sysfunc(time()),
e86011z.));
%put NOTE: ;
%put NOTE: == Modification History
=====;
%put NOTE: Date      Initials   No. Reason;
%put NOTE: 18Sep2014  JMH        1) Amended tranwrđ for param to
present units;
%put NOTE: 18Sep2014  JMH        2) Amended proc freq for clinical
significance;
%put NOTE: 18Sep2014  JMH        3) Amended to present change from
baseline correctly;
%put NOTE: 18Sep2014  JMH        4) Amended baseline footnote;
%put NOTE: 18Sep2014  JMH        5) Amended DPs for Protein;
%put NOTE: 18Sep2014  KB         6) Amended units mg/dl to mg/dL;
%put NOTE: ;
%put NOTE:
=====;
options notes source source2 nofullstimer validvarname=upcase missing='
';
ods _all_ close;
ods listing;

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

%let tflno=T_15_02_06_15(urine);

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

```

```

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 values for column headers*/
data adsl;
    set adam.adsl(where=(saffl='Y'));
    output;
    trt01an=99;
    trt01a='Overall Safety';
    output;
run;

proc sort data=adsl nodupkey out=adsl1;
    by trt01an trt01a subjid;
run;

proc freq data=adsl1(where=(not missing(trt01an))) noprint;
    table trt01an*trt01a/ out =tot(drop=percent rename=(count=total
trt01an=trtan trt01a=trta));
run;

data tot2;
    set tot;
    call symput('trt' || compress(put(trtan,best.)), compress(total));
run;

/* Urinalysis data */

data adlba;
    set adam.adlb(where=(saffl = 'Y' and parcat1='Urinalysis' and
(anl01fl='Y') and lbstat ne 'NOT DONE'));
    length paramc $132.;
    if missing(trtan) then delete;
    if index(trta,'Exposed') then delete;

    if paramn not in(201 202) and aval gt 0 then avalc='Positive';

    len=length(scan(param,2,''));
/*    param=tranwrd(param,'(mg/dL)', ''); */ /* 1) JMH 18Sep2014 */
    param=upcase(substr(param,1,1))||lowercase(substr(param,2));
    param=tranwrd(param,'g/l','g/L');
    param=tranwrd(param,'eryt/ul','ERYT/uL');
    PARAM=TRANWRD(PARAM,'mg/dl','mg/dL'); /* 6) KB 18Sep2014 */

    anrmiss=cmiss(anrhi,anrlo);

```

```

        if anrmiss=0 then paramc=strip(param)||'
$N'||strip(propcase(anrlo))||'-'||strip(propcase(anrhi));
        else if anrmiss>0 then do;
            paramc=strip(param)||'
$N'||strip(propcase(anrlo))||strip(propcase(anrhi));
        end;

        avisit=tranwrd(avisit,'/','/ ');
        output;
        trtan=99;
        trta='Overall Safety';
        output;
run;

data adlb;
    set adlba;

        if ablfl='Y' then do;
            avisit='Baseline';
            avisitn=100;
        end;
        if avisit ne 'Baseline' and avisitn le 99 then delete;
run;

/*Categorical data - Negative/Trace/Positive etc.*/
data cat_ntp;
    set adlb(where=(paramn>202));
    format stat $100.;
    stat=avalc;
run;

proc sort data=cat_ntp;
    by trtan trta paramn paramc paramcd avisitn avisit anrlo anrhi;
run;

proc freq data=cat_ntp noprint ;
    by trtan trta paramn paramc paramcd avisitn avisit anrlo anrhi;
    tables stat / out=cat_ntp2(drop=percent);
run;

proc sort data=cat_ntp2;
    by trtan trta;
run;

data percents1;
    merge cat_ntp2 tot2;
    by trtan trta;
    attrib value length=$200.;

    percent=count/total*100;

    if percent=100 then value=strip(put(count,8.)) || ' (100 %)';
    else if percent ge 10 then value=strip(put(count,8.)) || ' ( ' ||
strip(put(round(percent,0.1),8.1)) || '%)';

```

```

        else if percent lt 10 then value=strip(put(count,8.)) || ' ( ' ||
strip(put(round(percent,0.1),8.1)) || '%)';
        else put "WA" "RNING: Unexpected value for percent, please check "
trta= paramc= percent= ;
run;

proc sort data=percents1;
    by paramn paramc avisitn avisit stat;
run;

proc transpose data=percents1 out=tdatal prefix=t name=varname;
    by paramn paramc avisitn avisit stat;
    var value;
    id trtan;
    idlabel trta;
run;

/*Low/High/Abnormal counts*/
data cat_lha;
    set adlb;
    format stat $100.;
    stat=propcase(anrind);
run;

proc sort data=cat_lha;
    by trtan trta paramn paramc paramcd avisitn avisit anrlo anrhi
aclsig;
run;

proc freq data=cat_lha noprint ;
    by trtan trta paramn paramc paramcd avisitn avisit anrlo anrhi;
    tables stat / out=cat_lha2(drop=percent);
run;

/* 2) start JMH 18Sep2014 */
DATA CAT_LHA_TEST;
    SET CAT_LHA;
    WHERE ACLSIG='CS';
    STATDUM='Y';
RUN;
/* 2) end JMH 18Sep2014 */

/*Get counts for Abnormal - clinically significant*/
proc freq data=/*cat_lha*/CAT_LHA_TEST noprint; /* 2) JMH 18Sep2014 */
    by trtan trta paramn paramc paramcd avisitn avisit anrlo anrhi
aclsig;
    tables /*stat*/STATDUM / out=abnorm(where=(aclsig in('Y' 'CS'))
drop=percent); /* 2) JMH 18Sep2014 */
run;

data abnorm2;
    set abnorm;
    stat='Abnormal clinically significant';
    drop aclsig;

```

```

run;

data cat_lha3;
    set cat_lha2 abnorm2;
run;

proc sort data=cat_lha3;
    by trtan trta;
run;

data percents2;
    merge cat_lha3 tot2;
    by trtan trta;
    attrib value length=$200.;

    percent=count/total*100;

    if percent=100 then value=strip(put(count,8.)) || ' (100 %)';
    else if percent ge 10 then value=strip(put(count,8.)) || ' ( ' ||
strip(put(round(percent,0.1),8.1)) || '%)';
    else if percent lt 10 then value=strip(put(count,8.)) || ' ( ' ||
strip(put(round(percent,0.1),8.1)) || '%)';
    else put "WA" "RNING: Unexpected value for percent, please check "
trta= paramc= percent= ;
run;

proc sort data=percents2;
    by paramn paramc avisitn avisit stat;
run;

proc transpose data=percents2 out=tdata2 prefix=t name=varname;
    by paramn paramc avisitn avisit stat;
    var value;
    id trtan;
    idlabel trta;
run;

/*Merge all categorical data */
data cat;
    set tdata1(in=a) tdata2(in=b);
    if a then statord1=2;
    else if b then statord1=3;
    /*Negative, Trace and Positive. These will be presented before
Normal/Abnomral etc*/
    if stat='Negative' then statord2=1;
    else if stat='Trace' then statord2=2;
    else if stat='Positive' then statord2=12;

    /*Low/High/Normal/Abnormal/ Abnormal CS should be presented last*/
    else if stat='Low' then statord2=14;
    else if stat='Normal' then statord2=15;
    else if stat='High' then statord2=16;
    else if stat='Abnormal' then statord2=17;
    else if stat='Abnormal clinically significant' then statord2=18;

```

```

        /*Now assign orders for all other character results, these should
be output first*/
        else if stat='1+' then statord2=3;
        else if stat='2+' then statord2=4;
        else if stat='3+' then statord2=5;
        else if stat='Small' then statord2=6;
        else if stat='Moderate' then statord2=7;
        else if stat='Large' then statord2=8;

        else put "WA" "RNING: Assign statord2 for stat: " stat= paramc=
t99=;

        if statord2 in(14 15 16 17) then stat=strip(stat)||' value - n
(%)';
        else stat=strip(stat)||' - n (%)';

run;

proc sort data=cat; by paramn avisitn statord2; run;

/*Summary statistics*/
data stats;
    set adlb(where=(not missing(aval)));
    statval=aval;
    output;
    if avisitn ne /*104*/106 then delete; /* 3) JMH 18Sep2014 */
    avisitn=204;
    avisit='Change from baseline';
    statval=chg;
    output;
run;

proc sort data=stats;
    by trtan trta paramn paramc paramcd avisitn avisit anrlo anrhi;
run;

proc univariate data=stats noprint;
    var statval;
    by trtan trta paramn paramc paramcd avisitn avisit anrlo anrhi;
    output out=stats2 N=N1 mean=mean1 std=std1 median=med1 min=min1
max=max1;
run;

data stats3;
    set stats2;
    attrib meansd minmax n median length=$20.;

    n = left(compress(put(n1,8.)));
    if paramcd in ('PH') then do;
        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(put(0.001*ceil(std1/0.001),8.3)) || ')';
        if not missing(min1) and not missing(max1) then minmax =
left(compress(put(min1,8.1))) || ', ' || left(compress(put(max1,8.1)));
        end;
        else if paramcd in ('SPGRAV') then do;
            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)) || ')';
            if not missing(min1) and not missing(max1) then minmax =
left(compress(put(min1,8.))) || ', ' || left(compress(put(max1,8.)));
            end;
/* 5) start JMH 18Sep2014 */
        ELSE IF PARAMCD IN ('UPROT') THEN DO;
            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(PUT(0.0001*CEIL(STD1/0.0001),8.4)) || ')';
            IF NOT MISSING(MIN1) AND NOT MISSING(MAX1) THEN MINMAX =
LEFT(COMPRESS(PUT(MIN1,8.2))) || ', ' || LEFT(COMPRESS(PUT(MAX1,8.2)));
            END;
/* 5) end JMH 18Sep2014 */
        else do;
            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)) || ')';
            if not missing(min1) and not missing(max1) then minmax =
left(compress(put(min1,8.))) || ', ' || left(compress(put(max1,8.)));
            end;

            if n1<4 then do; median='NC'; meansd='NC'; minmax='NC'; end;

/*          if avisitn=204 and paramn not in(201 202) then delete; */ /*
4) JMH 18Sep2014 */

```

```

        drop n1 mean1 std1 med1 min1 max1;
run;

```

```

proc sort data=stats3;
    by paramn paramc avisitn avisit;
run;

```

```

proc transpose data=stats3 out=stats4 prefix=t name=varname;
    by paramn paramc avisitn avisit;
    var n meansd median minmax;
    id trtan;
    idlabel trta;
run;

```

```

data stats5;
    set stats4;
    by paramn paramc avisitn avisit;
    if not first.avisitn then statord2+1;
    else statord2=1;
run;

data zeroflags;
    set stats5(where=(varname='N'));
    array a[5] t1-t3 t97 t99;
    array fl[5] fl1-fl3 fl97 fl99;
    do i=1 to 5;
        if missing(a[i]) or a[i]='0' then fl[i]=1;
    end;
run;

data results;
    set cat stats5(in=a);
    format stat $100.;
    if a then do;
        statord1=1;
        if varname='N' then stat='n';
        else if varname='MEANSD' then stat='Mean (SD)';
        else if varname='MINMAX' then stat='Min, Max';
        else stat=propcase(varname);
    end;
run;

proc sort data=results;
    by paramn paramc avisitn avisit statord1 statord2;
run;

/*Create dummy rows - Each study will be different!!! Please check*/
proc sort data=results out=results_b(keep=paramc paramn avisit avisitn)
nodupkey;
    by avisitn paramn;
run;

data extra0;
    set results_b(where=(avisitn ne 204 and paramn not in(201 202)));
    format stat $100.;
    by avisitn paramn;

    statord1=2;
    stat='Negative - n (%)';
    statord2=1;
    output;
    stat='Positive - n (%)';
    statord2=12;
    output;
run;

data extral1;

```



```

set results_b(where=(avisitn ne 204 and paramn not in(201 202)));
format stat $100.;
by avisitn paramn;

    statord1=3;
    stat='Normal value - n (%)';
    statord2=15;
    output;
    stat='Abnormal value - n (%)';
    statord2=17;
    output;
    stat='Abnormal clinically significant - n (%)';
    statord2=18;
    output;
run;

data extra2;
    set results_b(where=(avisitn ne 204 and paramn in(201 202)));
    format stat $100.;
    by avisitn paramn;

        statord1=3;
        stat='Normal value - n (%)';
        statord2=15;
        output;
        stat='Low value - n (%)';
        statord2=14;
        output;
        stat='High value - n (%)';
        statord2=16;
        output;
        stat='Abnormal clinically significant - n (%)';
        statord2=18;
        output;
run;

data extra;
    set extra0 extra1 extra2;
run;

proc sort data=extra;
    by avisitn paramn statord1 statord2 stat;
run;

proc sort data=results;
    by avisitn paramn statord1 statord2 stat;
run;

data results2;
    merge results(drop = varname ) extra;
    by avisitn paramn statord1 statord2 stat;
run;

```

```

proc sort data=results2;
  by paramn paramc avisitn avisit;
run;

data results3;
  merge results2 zeroflags;
  by paramn paramc avisitn avisit;
run;

data results4;
  set results3;
  array t[5] t1-t3 t97 t99;
  array fl[5] fl1-fl3 fl97 fl99;
  do i=1 to 5;
    if fl[i]=1 then do;
      if stat='n' and missing(t[i]) then t[i]='0';
      else if statord2>1 and missing(t[i]) then t[i]='';
    end;
    if stat not in('Mean (SD)' 'Median' 'Min, Max') then do;
      if fl[i] ne 1 then do;
        if missing(t[i]) then t[i]='0';
      end;
    end;
    else if fl[i] eq 1 and missing(ncfl) then do;
      if missing(t[i]) then t[i]='0';
    end;
    end;
    if index(t[i], 'NC') then ncfl=1;
  end;

  paramc=tranwrd(paramc, 'Ph', 'pH');

  drop i varname;
run;

proc sort data=results4;
  by paramn avisitn statord1 statord2;
run;

proc sql noprint;
  create table table.T_15_02_06_15 as
  select paramc, avisit, stat, t1, t2, t3, t97, t99
  from results3
  order by paramn, avisitn, statord1, statord2;
quit;

data paging;
  set results4;
  by paramn avisitn statord1 statord2;

  flag=1;

  if first.avisitn then ln=1;
  else ln+1;
  if ln=1 then page+1;

```

```

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

options number nodate orientation=landscape papersize=&p_pgsz 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;

%macro outrtf(blankn=, halfblnk=);

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

ods path stdlib.tl06324 (read) ;
ods results off;
ods rtf toc_data/* contents*/
file="/cvn/projects/prj/data/000000106324/TFL/&TFL_Part./&tflno..rtf"
style=tl06324 startpage=yes headery=1440 footery=1440 ;
ods noproctitle;
%do i=1 %to &page;
ods proclabel = ' ';

title ;
footnote;
%let wd=0;
%let nc=0;
%let footnote=0;

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

    /* Amend title as needed */
    _firtitl="Table 15.2.6.15    Summary of Urinalysis Parameters -
Safety Population";
    _upcas=(length(_firtitl)-
length(compress(_firtitl,'ABCDEFGHIJKLMNOPQRSTUVWXYZ')))/2;
    len=&blankn.-length("(Page &i of &page)");
    if ncfl=1 then call symput('nc',1);
    if eof then do;
        call symput('_FSRTITL', trim(left(_firtitl)));
        call symput('_blankn', compress(put(len,best.)));
    end;
    drop _firtitl _upcas len;

    if paramn in (207 218) then call symput('footnote',1);
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 missing headline headskip nowd split = '#' %if
&i=1 %then %do; contents=' ' %end; %else %do; contents='' %end;;;
    column flag page ("Parameter (units)#Reference range" paramc)
avisitn ("Study Day" avisit) ("Statistic" stat)
    ("THS 2.2#(N=&trt1)" t1) ("CC#(N=&trt2)" t2) ("SA#(N=&trt3)" t3)
("Enrolled Not #Randomized#(N=&trt97)" t97) ("Overall#Safety#(N=&trt99)"
t99) ;

    define flag          / order order=internal noprint;
    define page          / order order = internal noprint;
    define avisitn       / order order = internal noprint;
    define paramc        / group style={just=l cellwidth=2.8cm}
style(header)={just=c} "";
    define avisit        / group style={just=l cellwidth=1.8cm}
style(header)={just=c} "";
    define stat          / style={just=l cellwidth=2.9cm}
style(header)={just=c} "";
    define t1-t3         / style={just=c cellwidth=1.7cm}
style(header)={just=c} "";
    define t97           / style={just=c cellwidth=1.9cm}
style(header)={just=c} "";
    define t99           / style={just=c cellwidth=1.7cm}
style(header)={just=c} "";

    break before flag / page %if &i=1 %then %do;
    contents="&_fsrtitl" %end; %else %do; contents='' %end;;

    break after page / page;

    compute before page / style={protectspecialchars=off};
        line "&linetop";
    endcomp;

    compute after avisit/ style={protectspecialchars=off};
        line "";
    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";
    endcomp;

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

```

```

        line 'Note: CC = Conventional cigarettes; SA = Smoking
abstinence; THS = Tobacco Heating System.';
        line "Note: Enrolled Not Randomized refers to all subjects enrolled
but not randomized. Overall Safety refers to enrolled subjects exposed to
THS 2.2.";
        line "Note: Percentages are based on the number of subjects
indicated in the column header (N).";
/*        line 'Note: Baseline is defined as the last assessment prior
to 06:29 AM on Day 1.'; */
        LINE 'Note: Baseline is the last assessment prior to
first product use in CC/THS 2.2 arms on Day 1 or last assessment prior to
06:29 AM in SA arm on Day 1.'; /* 4) JMH 18Sep2014 */
        %if &footnote=1 %then %do;
        line "Note: If result is greater than 'Negative/Trace' then
result is counted as 'Positive'.";
        %end;
        %if &nc=1 %then %do;
        line 'Note: NC = Not calculated.';
        %end;
        line "";
        line "Appendix 15.3.6.6";
        line "Path: &TFLpath." &_blankn.*"\~\~" "(Page &i of &page)";
        line "Program Run: &sysdate &sysuserid Program Status:
&status";
        endcomp;
run;
%end;
ods rtf close;
ods results on;
ods path sashelp.tmplmst (read);

%mend ;

%outrtf(blankn=70, halfblnk=N);
ods listing;
proc printto print = "&table./T_15_02_06_15.lst" new;
run;

proc contents data = table.T_15_02_06_15 varnum;
run;
ods listing close;

proc printto ; run;
*=====;
* END OF PROGRAM CODE ;
*=====;

```