

```

%_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_ecg.sas;
%put NOTE: Purpose              : table of ecg measurements;
%put NOTE: ;
%put NOTE: Input Data           : ADAM.ADEG ADAM.ADSL;
%put NOTE: Output               : t_15_2_6_17(eg);
%put NOTE: Macros Called        : _MPRINTTO;
%put NOTE: ;
%put NOTE: Programmed by        : cvn_jriley;
%put NOTE: Creation Date        : 2014-07-24;
%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: 04Aug2014   JR         1) Amended dummy;
%put NOTE: 15Sep2014   KB         2) Amended clinical significance;
%put NOTE: 18Sep2014   JR         3) Amended baseline footnote;
%put NOTE: ;
%put NOTE:
=====;
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_17(eg);

/* 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;
    where saffl = '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='THS 2.2';
    output;
    trt01an=2;
    trt01a='CC';
    output;
    trt01an=3;
    trt01a='SA';
    output;
    trt01an=97;
    trt01a='Enrolled not randomized';
    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(count));
run;

```

```

/*Bring in appropriate data from ADEG*/
data adeg;
    set adam.adeg;
    where saffl = 'Y' and anl01fl='Y';
    if missing(trtan) then delete;
    if index(trta,'Exposed') then delete;
    output;
    trtan=99;
    trta='Overall Safety';
    output;
run;

/*INTP only as this code will bring out the class variables. Other params
will be dealt with later*/

proc freq data=adeg(where=(paramcd='INTP')) noprint;
    table trtan*trta*avisitn*avisit*aval*avalc*paramn*param*EGCLSIG /
out =intp1(drop=percent);    /* 2) KB 15Sep2014 */
run;

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

    trtan=1;
    trta='THS 2.2';
    output;
    trtan=2;
    trta='CC';
    output;
    trtan=3;
    trta='SA';
    output;
    trtan=97;
    trta='Enrolled not randomized';
    output;
run;

data tot3;
    set tot2;
    trta=trt01a;
    trtan=trt01an;
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;
    if b and not a then do;
    count = 0;
    avisitn=1;
    avisit='Screening';

```

```

        avalc='Normal';
    end;
    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;
/*    avalc='Abnormal, CNR';*/
    AVALC='Abnormal'; /* 2) KB 15Sep2014 */
    EGCLSIG='NCS'; /* 2) KB 15Sep2014 */
    output;
/*    avalc='Abnormal, CR';*/
    AVALC='Abnormal'; /* 2) KB 15Sep2014 */
    EGCLSIG='CS'; /* 2) KB 15Sep2014 */
    output;
run;

proc sort data=dumrows;
    by trtan trta avisitn avisit avalc EGCLSIG; /* 2) KB 15Sep2014 */
run;

proc sort data=intp2;
    by trtan trta avisitn avisit avalc EGCLSIG; /* 2) KB 15Sep2014 */
run;

data intp3;
    merge intp2(in=a) dumrows(in=b);
    by trtan trta avisitn avisit avalc EGCLSIG; /* 2) KB 15Sep2014 */
    if a or b;
    attrib statval length=$100.
        count1 length=$3.;
    if a then data=1;
    if b then dummy=1;
    if b and not a then do;
        count=0;
        percent=0;
    end;
        attrib paramc length = $100.
            visit length = $100.
            stat length = $100.;
    paramc=strip(param);
    if trtan=97 then do;

```

```

        paramc=strip('Clinical Relevance Assessment');
        paramn=0;
end;

ord = avisitn;
visit = avisit;

if avalc='Normal' then do;
    stat='Normal - n (%)';
    statord=1;
end;
/*
else if avalc='Abnormal, CNR' then do;*/
ELSE IF AVALC='Abnormal' AND EGCLSIG='NCS' THEN DO; /* 2) KB
15Sep2014 */
/*
    stat='Abnormal non-clinically relevant - n (%)';*/
    STAT='Abnormal non-clinically significant - n (%)'; /*
2) KB 15Sep2014 */
    statord=2;
end;
/*
else if avalc='Abnormal, CR' then do;*/
ELSE IF AVALC='Abnormal' AND EGCLSIG='CS' THEN DO; /* 2) KB
15Sep2014 */
/*
    stat='Abnormal clinically relevant - n (%)';*/
    STAT='Abnormal clinically significant - n (%)'; /* 2) KB
15Sep2014 */
    statord=3;
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 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 = count1 || ' ' || '(
' ||left(strip(put(round(percent,0.1),5.1))) || '%)';
    else if percent ge 10 then statval = count1 || ' ' || '(
' ||left(strip(put(round(percent,0.1),5.1))) || '%)';
end;

run;

proc sort data=intp3;
    by paramn paramc ord visit avisitn statord stat;
run;

```

```

proc transpose data=intp3 out=intp4a(where=(stat ne 'DUMMY')) prefix=t;
    by paramn paramc ord visit avisitn statord stat;
    var statval;
    id trtan;
    idlabel trta;
run;

data intp4;
    set intp4a;

    if t1='' then t1='0';
    if t2='' then t2='0';
    if t3='' then t3='0';
    if t97='' then t97='0';
    if t99='' then t99='0';

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=106 then ord=2;
    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=2; /*Change from Baeline to Day 6*/
    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=nlo mean=meanlo std=stdlo median=medlo
min=minlo max=maxlo;
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 meansdo length=$100.
                                minmaxo length=$100.
                                no        length=$100.
                                mediano  length=$100.;

                                no        = left(compress(put(n1o,8.)));
                                if not missing(med1o) then mediano =
left(compress(put(med1o,8.1)));
                                if not missing(mean1o) and not missing(std1o) then
meansdo = left(compress(put(mean1o,8.1))) || ' (' ||
compress(put(0.01*ceil(std1o/0.01),8.2)) || ')';
                                if not missing(min1o) and not missing(max1o) then
minmaxo = left(compress(put(min1o,8.))) || ', ' ||
left(compress(put(max1o,8.)));

    attrib meansdc length=$100.
                                minmaxc length=$100.
                                nc        length=$100.
                                medianc  length=$100.;

                                nc        =left(compress(put(n1c,8.)));
                                if not missing(med1c) then medianc =
left(compress(put(med1c,8.1)));
                                if not missing(mean1c) and not missing(std1c) then
meansdc = left(compress(put(mean1c,8.1))) || ' (' ||
compress(put(0.01*ceil(std1c/0.01),8.2)) || ')';
                                if not missing(min1c) and not missing(max1c) then
minmaxc = left(compress(put(min1c,8.))) || ', ' ||
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) ';*//
    PARAMC='Heart Rate (Beats/min)'; /* 1) JR
04Aug2014 */
  end;
run;

proc sort data=results03;
by paramn paramc ord visit avisitn;
run;

proc transpose data=results03 out=results04_orig1 prefix=o
name=varname;
  by paramn paramc ord visit avisitn;
  var no meansdo mediano minmaxo;
  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 visit avisitn;
  var nc meansdc medianc minmaxc;
  id trtan;
  idlabel trta;
run;

proc sort data=results04_orig;
  by paramn paramc ord visit avisitn varname;
run;

```



```

proc sort data=results04_chg;
  by paramn paramc ord visit avisitn varname;
run;

data results04;
  merge results04_orig results04_chg;
  by paramn paramc ord visit avisitn varname;
run;

data results05;
  set results04;
  attrib stat length = $100.;
  if varname='NC' then do; statord=1; stat='n'; end;
  else if varname='MEANSDC' then do; statord=2; stat='Mean
(SD)'; end;
  else if varname='MEDIANC' then do; statord=3; stat='Median';
end;
  else if varname='MINMAXC' then do; statord=4; stat='Min,
Max'; end;

  drop varname;
run;

data results06;
  set results05;
  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(o97) then o97='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(o97) then o5='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(c97) then c97='0';
    if missing(c99) then c99='0';
  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(t97) then t97='0';
    if missing(t99) then t99='0';
  end;

  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=7; end;
  else put "WA" "RNING: Check parameter text is sentence case "
paramc= ;

  if index(visit,'Screening') then
visit=tranwrd(visit,'Screening','Baseline');

  flag=1;
run;

data labels;
  set allresults;
  attrib
    o1 o2 o3 o97 o99 label = "Raw value"
    c1 c2 c3 c97 c99 label = "Change"
    paramc label='Parameter (units)'
    visit label='Study Day'
    stat label='Statistic';

  if paramn = 7 then do;
    o1 = t1;
    o2 = t2;
    o3 = t3;
    o97 = t97;
    o99 = t99;
  end;

  visit=tranwrd(visit,'/', '/ ');
run;

proc sql noprint;

```

```

        create table table.T_15_02_06_17 as
        select paramc, visit, avisitn, stat, o1, c1, o2, c2, o3, c3, o97,
c97, o99, c99
        from labels
        order by paramn, ord, statord;
quit;

proc sort data=labels;
    by paramn ord statord;
run;

data paging;
    set labels;
    by paramn ord statord;
    /*if first.paramn and ln>6 then ln=1; *//*Amend to look
presentable, and avoid page overflows*/
    if first.paramn or (first.ord and ln ge 8) or (paramn=7 and ln gt
2) then ln=1;
    else ln+1;
    if ln=1 then page+1;
    call symput("page",compress(put(page,best.)));
run;

/* Standard - leave this */
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;
/* Standard - macro for paging */
%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
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;

title ;
footnote;
%let wd=0;
%let subpage=2;

%do j=1 %to &subpage;

```

```

ods proclabel = ' ';

%let maxpage=%eval(&page*&subpage);

%let npage=%eval(&subpage*&i+&j-&subpage);

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

    call symput('paramn',paramn);

    /* Amend title as needed */
    _firtitl="Table 15.2.6.17 Summary of ECG Measurements -
Safety Population";
    _upcas=(length("Path: &TFLpath.")-
length(compress("Path:&TFLpath.",'ABCDEFGHIJKLMNOPQRSTUVWXYZ')))/2;
    len=&blankn.-length("(page &npage of &maxpage)");
    if eof then do;
        call symput('_FSRTITL', trim(left(_firtitl)));
        call symput('_blankn', compress(put(len,best.)));
    end;
    drop _firtitl _upcas len;
run;

ods proclabel = ' ';
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 = '$' %if
&i=1 and &j=1 %then %do; contents=' ' %end; %else %do; contents='' %end;;
    column flag page paramn paramc ord visit statord stat %if &j=1
%then %do; ("THS 2.2 $(N=&trt1) &linebot" o1 c1) ("CC $(N=&trt2)
&linebot" o2 c2) ("SA $(N=&trt3) &linebot" o3 c3)%end;

    %else %if &j=2 %then %do; ("Enrolled not$randomized$(N=&trt97)
&linebot" o97 c97) ( "Overall$Safety$(N=&trt99) &linebot" o99
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=l cellwidth=1.7cm}
style(header)={just=c};
        define visit          / group style={just=l cellwidth=2cm}
style(header)={just=c};
        define stat          / group style={just=l cellwidth=1.6cm}
style(header)={just=c};
        %if &j=1 %then %do;
        define o1            / display style={just=c cellwidth=1.4cm}
style(header)={just=c};
        define c1            / display style={just=c cellwidth=1.4cm}
style(header)={just=c};
        define o2            / display style={just=c cellwidth=1.4cm}
style(header)={just=c};
        define c2            / display style={just=c cellwidth=1.4cm}
style(header)={just=c};
        define o3            / display style={just=c cellwidth=1.4cm}
style(header)={just=c};
        define c3            / display style={just=c cellwidth=1.4cm}
style(header)={just=c};
        %end;
        %else %if &j=2 %then %do;
        define o97           / display style={just=c cellwidth=1.4cm}
style(header)={just=c};
        define c97           / display style={just=c cellwidth=1.4cm}
style(header)={just=c};
        define o99           / display style={just=c cellwidth=1.4cm}
style(header)={just=c};
        define c99           / display style={just=c cellwidth=1.4cm}
style(header)={just=c};
        %end;

break before flag / page %if &i=1 and &j=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";
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: Change is change from baseline, where baseline is
defined as Screening.'; */
        line 'Note: Change is change from baseline, where 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.'; /* 3) JR
18Sep2014 */

        line ' ';
        line 'Appendix 15.3.6.9';
        line "Path: &TFLpath." &_blankn.*"\~\~" "(Page &npage of
&maxpage)";
        line "Program Run: &sysdate   &sysuserid   Program Status:
&status";
        endcomp;
run;
%end;
%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_17.lst" new;
run;

proc contents data = table.T_15_02_06_17 varnum;
run;
ods listing close;
proc printto ; run;
*=====;
* END OF PROGRAM CODE ;
*=====;

```