

```

%_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_bmi.sas;
%put NOTE: Purpose              : table of summary of weight and bmi;
%put NOTE: ;
%put NOTE: Input Data           : ADAM.ADSL ADAM.ADVS;
%put NOTE: Output               : t_15_2_6_19(bmi);
%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  KB        1) Added a warning;
%put NOTE: 04Aug2014  KB        2) Amended BMI category;
%put NOTE: 04Aug2014  KB        3) Amended casing of Enrolled Not
Randomized;
%put NOTE: 18Sep2014  JMH        4) Included BMI classifications for
Day 6;
%put NOTE: 18Sep2014  JR        5) Updated baseline footnoted;
%put NOTE: 17Oct2014  KB        6) Updated to show change from baseline
for BMI;
%put NOTE: 17Oct2014  KB        7) Amended paging from update 6;
%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_19(bmi);

/* 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 ;
*****;
/*Obtaining treatments*/
data dummytrts;
    attrib trta length=$40.
           trtan length=8.;

    trta="THS 2.2";
    trtan=1;
    output;
    trta="CC";
    trtan=2;
    output;
    trta="SA";
    trtan=3;
    output;
    trta="Enrolled not randomized";
    trtan=97;
    output;
    trta="Overall Safety";
    trtan=99;
    output;

run;

proc sort data=adam.adsl(where = (saffl = 'Y' and enrlfl = 'Y'))
out=adsl;
    by subjidn;
run;

data adsl2a;
    set adsl;
    if index(trt01a,'Exposed') or missing(trt01a)then delete;
    output;
    trt01a='Overall Safety';
    trt01an=99;
    output;
    keep subjidn trt01a trt01an ;
run;

proc freq data=adsl2a;

```

```

        table trt01a*trt01an/ noprint out=trt(drop=percent
rename=(trt01an=trtan trt01a=trta));
run;

proc sort data=trt;
    by trtan;
run;

data trt1;
    merge dummytrts(in=b) trt(in=a);
    by trtan;
    if a or b;
    if b and not a then count=0;
    call symput('trt' || compress(put(trtan,best.)), compress(count));
run;

/*Start of code for BMI classifications*/

data adsl_bmi1;
    set /*adsl*/ ADAM.ADVS(RENAME=(TRTA=TRT01A TRTAN=TRT01AN)); /* 4)
JMH 18Sep2014 */
    WHERE PARAMCD='DBMI'; /*Screenin data is not needed for the table*/
/* 4) JMH 18Sep2014 */
    IF ABLFL='Y' THEN AVISIT='Baseline'; /* 4) JMH 18Sep2014 */
    if index(trt01a,'Exposed') or missing(trt01a)then delete;
    output;
    trt01a='Overall Safety';
    trt01an=99;
    output;
run;

proc sort data=adsl_bmi1;
    by trt01an trt01a /*bmigr1n bmigr1*/ AVISITN AVISIT AVALCAT1; /* 4)
JMH 18Sep2014 */
run;

proc sort data=adsl_bmi1 nodupkey out=bmi3b(keep=trt01an trt01a AVISITN
AVISIT); /* 4) JMH 18Sep2014 */
    by trt01an trt01a AVISITN AVISIT; /* 4) JMH 18Sep2014 */
run;

data bmi3b_x;
    set bmi3b;
/*    bmigr1 = 'Underweight';*/ /* 4) JMH 18Sep2014 */
/*    bmigr1n=1;*/ /* 4) JMH 18Sep2014 */
    AVALCAT1='Underweight'; /* 4) JMH 18Sep2014 */
    dumflag=1;
    count=0;
run;

data adsl_bmi;
    merge adsl_bmi1(in=a) bmi3b_x(in=b);
    by trt01an trt01a /*bmigr1n bmigr1*/AVISITN AVISIT AVALCAT1; /* 4)
JMH 18Sep2014 */

```

```

        if a or b;
        rename trt01an=trtan trt01a=trta;
run;

data trt1_a;
    set trt1;
    rename count = total;
run;

proc freq data=adsl_bmi noprint;
    table trtan*trta/*bmigr1n*bmigr1*/AVISITN*AVISIT*AVALCAT1*dumflag
/ out =bmil(drop=percent); /* 4) JMH 18Sep2014 */
run;

data bmi2;
    merge bmil(in=a) trt1_a;
    by trtan trta;
    if a;
    percent=count/total*100; /*This works out the percentages*/

run;

data bmi3;
    set bmi2;
    attrib avalc avalc2 format=$20.
        stat format=$100.;

        BMIGR1=AVALCAT1; /* 4) JMH 18Sep2014 */

    paramn=16;
    paramcd='BMI (kg/m${super 2})';
/*    avisitn=99;*/
/*    avisit='Day -2'; */

    if bmigr1='Underweight' then do; stat='Underweight
(<18.5kg/m${super 2}) - n (%)'; order2=2000; end;
    else if bmigr1='Normal weight' then do; stat='Normal range (>=18.5
and <25kg/m${super 2}) - n (%)'; order2=2001; end;
    else if bmigr1='Overweight' then do; stat='Overweight (>=25 and
<30kg/m${super 2}) - n (%)'; order2=2002; end;
/*    else if bmigr1='Obese' then do; stat='Obese (>=25 and
<30kg/m${super 2}) - n (%)'; order2=2003; end;*/
    ELSE IF BMIGR1='Obese' THEN DO; STAT='Obese (>=30kg) - n (%)';
order2=2003; end; /* 2) KB 04Aug2014 */
    else if missing(bmigr1) then do; stat='Missing n (%)'; order2=2004;
end;
    ELSE PUT "USER WA" "RNING: Check BMIGR1s!" BMIGR1=; /* 1) KB
04Aug2014 */

    if not missing(bmigr1) then stat=left(trim(bmigr1)) || ' - n (%)';

    attrib          stat percentc length = $100.

```

```

statval statval2 length = $100.;

countc=left(compress(put(count,8.)));

if count ge 10 and count lt 100 then count1=compress(countc);
else if count lt 10 then count1=' '||compress(countc);

percentc=left(compress(put(round(percent,0.1),5.1))) || '%';

if percent=100 then percent1='(100 %)';
else if percent ge 10 then percent1='( ' ||
compress(percentc) ||')';
else if percent lt 10 then percent1='( ' ||
compress(percentc) ||')';

statval=trim(count1);
statval2=percent1;

if dumflag ne 1 and order2 not in(2000 2001 2002 2003 2004) then
do;
    avalc = strip(put(count,best.)) ;
    avalc2='(' ||strip(put(round(percent,0.1),5.1)) || ')';
end;

if dumflag ne 1 and order2 in (2000 2001 2002 2003 2004) then do;
    avalc = compress(count1) ;
    avalc2=percent1;
end;

else avalc='0';
run;

proc sort data=bmi3;
    by paramn paramcd avisitn avisit order2 stat;
run;

proc transpose data=bmi3 out=bmi4 prefix=t;
    by paramn paramcd avisitn avisit order2 stat;
    id trtan;
    idlabel trta;
    var avalc;
run;

proc transpose data=bmi3 out=bmi4p prefix=p;
    by paramn paramcd avisitn avisit order2 stat;
    id trtan;
    idlabel trta;
    var avalc2;
run;

data bmi4all;
    merge bmi4 bmi4p;
    by paramn paramcd avisitn avisit order2 stat;

```

```

        drop _name_;
run;

/*End of BMI classes. Set on with rest of stats later*/

/* All other data */
proc sort data=adam.advs(where = (saffl = 'Y' and enrfl = 'Y' and
anl01fl='Y' and paramcd in('BMI' 'WEIGHT' 'HEIGHT' 'DBMI')) out=advs;
    by subjidn paramcd avisit;
run;

/* These are all the variables required from ADVS */
data advs2_a;
    set advs;
    if index(trta,'Exposed') or missing(trta)then delete;
    if paramn ne 22 then do;
        paramcd1 = left(trim(paramcd))||'
(|||left(compress(avalu))|||)';
    end;
    else do;
        paramn=/*16*/15.5; /* 4) JMH 18Sep2014 */
        paramcd1 = left(trim(paramcd))||'${super 1}
(|||left(compress(avalu))|||)';
        paramcd1=tranwrd(paramcd1,'DBMI','BMI');
    END;
    output;
    trta='Overall Safety';
    trtan=99;
    output;

        drop paramcd;
run;

data advs_orig;
    set advs2_a;
run;

data advs_test;
    set advs2_a;
    drop aval avisit;
run;

data advs_chg;
    set advs_test(where=(avisitn in(106)));
    attrib avisit length=$40.;
    if avisitn=106 then avisitn=401;/*change from baeline to
discharge*/
    aval=chg;
/*    avisit='Change from Day -2 at Day 6'; */
    AVISIT='Change from Baseline at Day 6'; /* 4) JMH 18Sep2014 */
run;

data advs2;

```

```

set advs_orig advs_chg;
paramcd=paramcd1;

if paramcd='HEIGHT (cm)' then do;
    aval=aval/100;
    avalc=left(trim(put(aval,best.)));
    avalu='m';
    paramcd='HEIGHT (m)';
END;

keep usubjid paramcd param paramn trta trtan avisit avisitn avalc aval
valu;
run;

proc sort data=advs2;
    by usubjid;
run;

data advs3;
    merge advs2(in=a) adsl;
    by usubjid;
    if a;
    keep trtan trta paramn paramcd param avisitn avisit avalc aval
valu ;
run;

proc sort data=advs3;
    by trtan trta paramn paramcd param avisitn avisit avalc aval valu;
run;

%macro Stats(visit=, set=, stats=);
    proc means data=advs3 noprint;
        by trtan trta paramn paramcd param avisitn avisit;
        var aval;
        where avisit=&visit;
        output out=&stats;
        output out=&set median=;
    run;

%mend stats;

%stats(visit='Screening', set=scrn, stats=st1);
%stats(visit='Day -2', set=admi, stats=st2);
%stats(visit='Day 6/Discharge', set=disc, stats=st3);
%stats(visit=/'Change from Day -2 at Day 6'/'Change from Baseline at
Day 6';, set=chg, stats=st4); /* 6) KB 17Oct2014 */

data all;
    set scrn st1 admi st2 disc st3 chg st4;

    if missing(_stat_) then _stat_ = 'Median';

```

```

if _stat_='N' then order2=1001;
if _stat_='MEAN' then order2=1002;
if _stat_='STD' then order2=1003;
if _stat_='Median' then order2=1004;
if _stat_='MIN' then order2=1005;
if _stat_='MAX' then order2=1006;

attrib stat format=$100.
      avalc length=$20.;
if _stat_ eq 'STD' then stat='SD';
else stat=propcase(_stat_);
drop _stat_;

if paramcd ne 'HEIGHT (m)' then do;
    if stat='N' then avalc=left(trim(put((round(aval,1)),8.)));
    else if stat='SD' then avalc=
compress(put(0.001*ceil(aval/0.001),8.3));
    else if stat in('Mean' 'Median') then
avalc=left(trim(put((round(aval,0.01)),8.2)));
    else avalc=left(trim(put((round(aval,0.01)),8.1)));
end;
else do;
    if stat='N' then avalc=left(trim(put((round(aval,1)),8.)));
    else if stat='SD' then
avalc=compress(put(0.0001*ceil(aval/0.0001),8.4));
    else if stat in('Mean' 'Median') then
avalc=left(trim(put((round(aval,0.001)),8.3)));
    else avalc=left(trim(put((round(aval,0.001)),8.2)));
end;

run;

data meansd;
    set all (keep=trtan trta stat avalc paramcd paramn order2 avisitn
avisit);
    where stat in('Mean' 'SD'); /*Only keep mean and SD as this is all
we want here*/
run;

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

proc transpose data=meansd out=meansd1 prefix=m;
    by trtan trta paramn paramcd avisitn avisit;
    id order2;
    idlabel stat;
    var avalc;
run;

data meansd2;
    format stat $20. avalc $20.;
    set meansd1;

```



```

        if not missing(m1002) and not missing(m1003) then do;
            avalc=left(compress(m1002)) || ' (' ||
left(compress(m1003)) || ')';
        end;
        else if not missing(m1002) and missing(m1003) then do;
            avalc=left(compress(m1002));
        end;
        else if missing(m1002) and missing(m1003) then do;
            avalc='';
        end;
        stat='Mean (SD)';
        order2=1002;
        drop m1002 m1003;
run;
/*End of combining mean and SD*/

/*Combine Min and Max*/
data minmax;
    set all(keep=trtan trta stat order2 avalc paramn paramcd
avisitn avisit);
    where stat in('Min' 'Max'); /*Only keep min and max as this
is all we want here*/
run;

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

proc transpose data=minmax out=minmax1 prefix=m;
    by trtan trta paramn paramcd avisitn avisit;
    id order2;
    idlabel stat;
    var avalc;
run;

data minmax2;
    format stat $20. avalc $20.;
    set minmax1;
    if not missing(m1005) and not missing(m1006) then do;
        avalc=left(compress(m1005)) || ', ' ||
left(compress(m1006));
    end;
    else if missing(m1005) and missing(m1006) then do;
        avalc='';
    end;
    stat='Min, Max';
    order2=1005;
    drop m1005 m1006;
run;
/*End of combining min and max*/

/*Combine the new datasets which have Min, Max and Mean (SD)*/
data minmaxmeansd;
    set minmax2 meansd2;

```

```

run;

data all2;
    set all(where=(order2 ne 1002 and order2 ne 1003 and order2
ne 1005 and order2 ne 1006)) minmaxmeansd;
run;

data all3;
    set all2;
    if paramn=14 then paramn=17;
run;

proc sort data=all3;
    by trtan trta;
run;

data all4;
    merge all3(in=a) dummytrts(in=b);
    by trtan trta;
    if a or b;

    if b and not a then do;
        avalc = '';
        paramn=15;
        param='Weight';
        paramcd='WEIGHT (kg)';
        avisitn=1;
        avisit='Screening';
        order2=1001;
        stat='N';
    end;

    if paramn=16 and paramcd ne 'BMI${super 1} (kg/m2)' then
paramcd='BMI${super 1} (kg/m${super 2})';
    else if paramn=16 and paramcd='BMI${super 1} (kg/m2)' then
paramcd='BMI${super 1} (kg/m${super 2})';
run;

proc sort data=all4;
    by paramn paramcd avisitn avisit order2 stat ;
run;

proc transpose data= all4 prefix=t out=all5;
    by paramn paramcd avisitn avisit order2 stat;
    var avalc;
    id trtan;
    idlabel trta;
run;

data all6;
    set all5 bmi4all;
    format t1a t2a t3a t97a t99a $20.;

```

```

if stat = 'N' or order2 >=2000 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';
end;

paramcd=tranwrd(paramcd,'WEIGHT','Weight');
paramcd=tranwrd(paramcd,'HEIGHT','Height');
if stat = 'N' then stat = 'n';

flag=1;

IF index(AVISIT,'/') then avisit=tranwrd(avisit,'/',' ');
if avisit='Day -2' then avisit='Baseline';
if avisit=/'Change from Day -2 at Day 6'/'Change from Baseline at
Day 6' then avisit='Change from Baseline'; /* 6) KB 17Oct2014 */

if paramn in(15 16) then do;
  if avisitn in(1 401) then delete;
end;

array a[5] t1 t2 t3 t97 t99;
array b[5] p1 p2 p3 p97 p99;
array c[5] t1a t2a t3a t97a t99a;

do i=1 to 5;
  c[i]=strip(a[i])||' '||strip(b[i]);
end;
drop t1 t2 t3 t97 t99;
rename t1a=t1 t2a=t2 t3a=t3 t97a=t97 t99a=t99;
run;

proc sql noprint;
  create table table.t_15_02_06_19 as
  select paramcd, avisit, stat, t1, t2, t3, t97, t99
  from all6
  order by paramn, avisitn, order2;
quit;

proc sort data=all6;
  by paramn avisitn order2;
run;

data paging;
  set all6;
  by paramn avisitn order2;
/* if (paramn=16 and order2=2000) or (paramn=16 and avisitn=104 and
order2=1001) then ln=1;*/
  IF FIRST.PARAMN OR (FIRST.AVISITN AND LN GT 6) THEN LN=1; /* 4)
JMH 18Sep2014 */ /* 7) KB 17Oct2014 */
  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;
%let linebot2 = \brdrb\brdrs\brdrw15;

/* 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/* 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;

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

data comp;
    set paging end=eof;
    by paramn avisitn order2;
    where page=&i;

    /* Amend title as needed */
    _firtitl="Table 15.2.6.19 Summary of Weight and BMI Measurements -
Safety Population";
    _upcas=(length(_firtitl)-
length(compress(_firtitl,'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.)));
    end;
    call symput('ord',order2);
    drop _firtitl _upcas len;
run;

```

```

* 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 headsip missing nowd split = '#' %if
&i=1 %then %do; contents=' ' %end; %else %do; contents='' %end;;
      column flag page paramn ("Parameter (units)" paramcd) avisitn
("Study Day" avisit) order2 ("Statistic" stat )
      ("THS 2.2#(N=&trt1)" t1)
      ("CC#(N=&trt2)" t2)
      ("SA#(N=&trt3)" t3)
/*      ("Enrolled not#randomized#(N=&trt97)" t97)*/
      ("Enrolled Not#Randomized#(N=&trt97)" T97) /* 3) KB 04Aug2014
*/
      ("Overall#Safety#(N=&trt99)" t99);

      define flag          / order order = internal noprint;
      define page          / order order = internal noprint;
      define paramn        / order order=internal noprint;
      define paramcd       / order style={just=left
cellwidth=3.0cm}style(header)={just=center} "";
      define avisitn       / order order=internal noprint;
      define avisit        / group style={just=left
cellwidth=2cm}style(header)={just=center} "";
      define order2        / order order=internal noprint;
      define stat          / display style={just=left
cellwidth=3cm}style(header)={just=center} "";
      define t1            / display style={just=c
cellwidth=2.2cm}style(header)={just=center} "";
      define t2            / display style={just=c
cellwidth=2.2cm}style(header)={just=center} "";
      define t3            / display style={just=c
cellwidth=2.2cm}style(header)={just=center} "";
      define t97           / display style={just=center
cellwidth=2.3cm}style(header)={just=center} "";
      define t99           / display style={just=c
cellwidth=2.2cm}style(header)={just=center} "";

      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 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 avisit;
        line " ";
    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
who were 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
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.'; /* 5) JR
18Sep2014 */
/*        line 'Note: Change is change from baseline, where baseline is
defined as the last assessment prior to 06:29 AM on Day 1.';*/
        line 'Note: 1: BMI is derived in analysis datasets.';

        line "";
        line "Appendix 15.3.6.10";
        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_19.lst" new;
run;

proc contents data = table.T_15_02_06_19 varnum;
run;
ods listing close;
proc printto ; run;
*=====;
* END OF PROGRAM CODE ;
*=====;

```