

```

/*=====
| Covance Study Number      : 000000106343
| Program Name              : t_medhis.sas
| Purpose                   : Create table on medical history
| Input Data                : ADAM.ADMH ADAM.ADSL
| Output Data               : t_15_2_1_6(mh)
| Macros Called             :%m_printto , %m_logchk
| Originally Performed by   : Upender
| Date                     : 20Apr2015
|=====
| Modification History : Original Version
|-----
| Modified by          :
| Modification Date    :
| Modification Reason   :
+=====*/

```

```

%m_printto(route=YES);
options notes nosource;
proc datasets lib=work nolist memtype=data kill; quit;

```

```

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

```

```

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

```

```

%let tflno=T_15_02_01_06;

```

```

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

```

```

run;

```

```

*****;
* read in data ;
*****;

```

```

/*Use ADSL for column headers*/

```

```

data adsl;
    set adam.adsl;
    where safaf1 = 'Y';
    if missing(trt01a) or trt01a='Screen failure' then delete;
run;

```

```

data adsl_no_rand;
    set adam.adsl;
    where randf1 ne 'Y' and /*safbf1*/safbf1='Y';
    if missing(trt01a) or trt01a='Screen failure' then delete;
run;

```

```

data adsl1;
    set adsl adsl_no_rand;
    attrib headtext1 length =$200.
        headorder1 length=8.;
    headorder1=trt01an;
    headtext1=trt01a;
    output;
    trt01an=99;
    headorder1=99;
    trt01a='Overall Safety';
    headtext1='Overall Safety';
    output;
run;

```

```

data dumtrts; /*Use this to output any columns for which N=0*/
    attrib headtext1 length =$200.

```

```

        headorder1 length=8.;
headorder1=4;
headtext1='THSm2.2';
output;
headorder1=5;
headtext1='mCC';
output;
headorder1=3;
headtext1='SA';
output;
headorder1=96;
headtext1='Product Test';
output;
headorder1=99;
headtext1='Overall Safety';
output;
run;

proc sort data=dumtrts; by headorder1 headtext1; run;

proc freq data=ads11 noprint;
  table headorder1*headtext1/ out =tot(drop=percent);
run;

data tot2;
  merge tot(in=a) dumtrts(in=b);
  by headorder1 headtext1;
  if b and not a then count=0;
  call symput('trt' || compress(put(headorder1,best.)), compress(put(count,best.)));
run;

/*Bring in ADMH*/

proc sort data=adam.admh out=admh (where=( anymhf1='Y' and mhcat='MEDICAL HISTORY'));
  by usubjid;
run;

proc sort data=ads11 out=ads12 (keep=usubjid) nodupkey ; by usubjid; run;

data admh2 (rename=(trta=trt01a trtan=trt01an));
merge admh (in=a) ads12 (in=b);
by usubjid;
if a and b;
run;

data admh3;
set admh2;
attrib headtext1 length =$200.
        headorder1 length=8.;
headorder1=trt01an;
headtext1=trt01a;
output;
trt01an=99;
headorder1=99;
trt01a='Overall Safety';
headtext1='Overall Safety';
output;
run;

proc sort data=admh3; by headorder1 headtext1; run;

* Create an additional observation with missing VOL value for each table section;
* This is used to ensure that all table rows are output, even for rows with no adverse events;

data admh4;
  set admh3;
  output;
  mhbodsys='Any medical history';
  output;
run;

```

```

/** Number of CD's overall **/
proc freq data=admh4 noprint;
    tables headorder1*headtext1*mhbodsys / out=ovall(rename=(count=tot) drop=percent);
run;

/** getting number of subjects studied ****/
proc sort data=admh4 out=nmh3 nodupkey;
    by headorder1 headtext1 mhbodsys usubjid;
run;

proc freq data=nmh3 noprint;
    tables headorder1*headtext1*mhbodsys / out=novall(rename=(count=ntot) drop=percent);
run;

/*Merge on total number of events and subjects*/
data overall;
    merge ovall(in=a) novall(in=b);
    by headorder1 headtext1 mhbodsys;
/*Sort2 will sort the body systems, as we want Any concomitant disease to be the first row in the table,*/
/*we set sort2 to=1 for Any Adverse Events. For all other Body systems, it =2*/
if mhbodsys='Any medical history' then sort2=1;
else sort2=2;
run;

/* Ordering the body systems with most frequent first*/

/*Body systems ordered by highest number of events*/
proc freq data=admh4 noprint;
    tables headorder1*headtext1*mhbodsys / out=tmhbod(rename=(count=tot) drop=percent);
run;

/* Body systems ordered by most subjects */
proc sort data=admh4 out=nmh2 nodupkey;
    by headorder1 headtext1 mhbodsys subjidn;
run;

proc freq data=nmh2 noprint;
    tables headorder1*headtext1*mhbodsys / out=mhbod(rename=(count=ntot) drop=percent);
run;

/*Combine number of events and subjects for each body system*/
data body;
    merge tmhbod mhbod;
    by headorder1 headtext1 mhbodsys;
run;

data body2;
    set body;
/*Sort 2 sorts body system, sort3 will sort preferred term, so here sort3 can=0 as preferred term isnt included yet*/
if mhbodsys='Any medical history' then do; sort2=1; sort3=0; end;
else do; sort2=2; sort3=0; end;
run;

/* Sorting bodysystem by preferred term */

/*Sorting events*/
proc freq data=admh4 noprint;
    tables headorder1*headtext1*mhbodsys*mhdecod / out=preft(rename=(count=tot) drop=percent);
run;

/*Sorting subjects*/
proc sort data=admh4 out=npmh2 nodupkey;
    by headorder1 headtext1 mhbodsys mhdecod subjidn;
run;

proc freq data=npmh2 noprint;
    tables headorder1*headtext1*mhbodsys*mhdecod / out=npref(rename=(count=ntot) drop=percent);
run;

/*Combining events and subjects for each preferred term within each body system*/
data prefterm;
    merge preft npref;
    by headorder1 headtext1 mhbodsys mhdecod;
run;

data mhdecod;
    set prefterm;

```

```

/*Sort 2 sorts body system, sort3 will sort preferred term.*/
/*Here sort3 orders the preferred terms by most events then most subjects, within each body system*/
if mhbodsys='Any medical history' then do; sort2=1; sort3=0; end;
    else do; sort2=2; sort3=1; end;
run;

/*Combine all data together*/
data all;
    set overall body2 mhdecod;
run;

proc sort data=all out=all2;
    by headorder1 headtext1 sort2 mhbodsys mhdecod;
run;

/*Merge on the totals from ADSL to work out percentages. Include dummy for Sequences with no AEs.*/
data format;
    merge all2(in=a) dumtrts tot;
    by headorder1 headtext1;
if not a then do;
    sort2=1;
    sort3=0;
    mhbodsys='Any medical history';
    dumflag=1;
end;
run;

data format2;
    set format;
    attrib text text2 text3 format=$20.;
    /* Percentage of subjects*/
    if not missing(count) then percent=put((ntot/count)*100,8.1);
    else percent='0';

    /* display variables*/

/*n value*/
if missing(ntot) then text='0';
else text=put(ntot,3.);

/*% value*/
if missing(percent) then text3='';
else if percent="100" then text3='(100 )';
else if percent ge "10" then text3='( ||compress(put(percent,8.1))||)';
else if percent lt "10" then text3='( ||compress(put(percent,8.1))||)';

/*events value*/
if missing(tot) then text2='';
    else text2=compress(put(tot,3.));

/*Any concomitant disease isn't broken down into bodsystems or preferred term so some rows can be deleted*/
if dumflag ne 1 and mhbodsys='Any medical history' and (not missing(mhdecod) or missing(sort3)) then delete;
drop percent dumflag;
run;

proc sort data=format2; by headorder1 headtext1 sort2 mhbodsys sort3 mhdecod; run;

proc sort data=format2 out=format3; by sort2 mhbodsys sort3 mhdecod; run;

/*Transpose n values*/
proc transpose data=format3 out=nformat (drop=_name_) prefix=n;
    by sort2 mhbodsys sort3 mhdecod;
    var text;
    id headorder1;
    idlabel headtext1;
run;

/*Transpose % values*/
proc transpose data=format3 out=performat (drop=_name_) prefix=p;
    by sort2 mhbodsys sort3 mhdecod;
    var text3;
    id headorder1;
    idlabel headtext1;
run;

/*Transpose event values*/
proc transpose data=format3 out=eformat (drop=_name_) prefix=e;

```

```

    by sort2 mhbodsys sort3 mhdecod;
    var text2;
    id headorder1;
    idlabel headtext1;
run;

/*Combine all the transposed datasets to get n, % and events columns in one dataset*/
data tformat;
merge nformat eformat performat;
by sort2 mhbodsys sort3 mhdecod;

if missing(sort3) then delete;
run;

/*Order bodysystems by number of events and subjects*/
data bodsyssort;
set tformat;
where sort3=0;
/*Create numeric variables for sorting*/
e99_n=input(e99,8.);
n99_n=input(n99,8.);
run;

proc sort data=bodsyssort;
by sort2 sort3 descending n99_n descending e99_n ;
run;

data bodsyssort2;
set bodsyssort;
odd+1;
keep sort2 mhbodsys odd;
run;

/*Merge this back onto tformat to include the variable odd which orders the bodysystems*/
proc sort data=tformat; by sort2 mhbodsys sort3 mhdecod; run;
proc sort data=bodsyssort2; by sort2 mhbodsys; run;

data systems;
merge tformat bodsyssort2;
by sort2 mhbodsys ;
run;

/*Order preferred terms within each bodysystem by number of events and subjects*/
data prefsort;
set systems;
where sort3=1;
/*Create numeric variables for sorting*/
e99_n=input(e99,8.);
n99_n=input(n99,8.);
run;

proc sort data=prefsort;
by odd descending n99_n descending e99_n;
run;

data prefsort2;
set prefsort;
odd2+1;
keep sort2 mhbodsys sort3 mhdecod odd odd2;
run;

/*Merge this back onto systems to include the variable odd2 which orders the preferred terms*/
proc sort data=systems; by sort2 mhbodsys sort3 mhdecod; run;
proc sort data=prefsort2; by sort2 mhbodsys sort3 mhdecod; run;

data final;
merge systems (drop=odd) prefsort2 (drop=odd);
by sort2 mhbodsys sort3 mhdecod;
run;

data final_z;
merge final bodsyssort2 (drop=sort2);
by mhbodsys;
run;

/*Now the data can be sorted using odd and odd2*/

```

```

proc sort data=final_z;
  by sort2 odd odd2;
run;

data final2;
  set final_z;
  attrib column format=$200. label='Formatted text'
         column2 format=$200. label='Unformatted text';

attrib wrap length = $200;
wrap = mhdecod;

i=25; *This is the max length allowed on a single line - change as needed;
if length(wrap)>i then do;
  nwraps = ceil(length(wrap)/i); *Calculate how many lines the text will wrap over;
do while(nwraps > 0);
  fin=0;
  j = min(length(mhdecod),i*nwraps); *Calculate starting point - loop will cycle backwards from this point looking for a space;

  K=J;
do while(fin=0 and j gt ((i*(nwraps-1))+1));
  if j = k and substr(wrap,j,1)=' ' then wrap = substr(wrap,1,j-1) || " |n |S={foreground=white} . |S={} " || substr(wrap,j+1);

  if substr(wrap,j,1)=' ' and j ne k then do;
    wrap=substr(wrap,1,j-1) || " |n |S={foreground=white} . |S={} " || substr(wrap,j+1);
    fin=1;
  end;
  else j=j-1; *No space found - move back one character;
end;
nwraps=nwraps-1; *Once this wrap is handled, move up a line until all are handled (when nwraps = 0);
end;
end;

if sort3=1 then column2 = mhdecod;
if sort2=2 and sort3=0 then column2=left(trim(mhbodsys));
if sort2=1 then do; column2='Any medical history'; odd2=0; end;

if sort3=1 then column = "|S={foreground=white} . |S={} " || wrap ;
if sort2=2 and sort3=0 then column=left(trim(mhbodsys));
if sort2=1 then do; column='Any medical history'; odd2=0; end;

if column='|S={foreground=white} . |S={} GASTROOESOPHAGEAL REFLUX|n |S={foreground=white} . |S={} DISEASE' then do;
  column='|S={foreground=white} . |S={} GASTROOESOPHAGEAL|n |S={foreground=white} . |S={} REFLUX DISEASE';
end;

run;

proc sort data=final2;
  by sort2 odd odd2;
run;

data labels;
  set final2;
  attrib n3 label = " n"
         n4 label = " n"
         n5 label = " n"
         n96 label = " n"
         n99 label = " n"
         p3 label = '(%)'
         p4 label = '(%)'
         p5 label = '(%)'
         p96 label = '(%)'
         p99 label = '(%)'
         e3 label = "Events"
         e4 label = "Events"
         e5 label = "Events"
         e96 label = "Events"
         e99 label = "Events";
run;

data final4;

```

```

        set labels;

        array a [4] n3 n4 n5 n96;
        do i=1 to 4;
            if missing(a[i]) then a[i] ='0';
        end;

        if missing(n99) then n99='0';

        flag=1;

run;

data final4a (keep=odd flagx flg_n3 flg_n4 flg_n5 flg_n96);
set final4;
flagx='Y';
if sort3=0 and (strip(n3)='0' or strip(n4)='0' or strip(n5)='0' or strip(n96)='0') then do;
if strip(n3)='0' then flg_n3='Y';
if strip(n4)='0' then flg_n4='Y';
if strip(n5)='0' then flg_n5='Y';
if strip(n96)='0' then flg_n96='Y';
end;
if sort3=0 and (strip(n3)='0' or strip(n4)='0' or strip(n5)='0' or strip(n96)='0') then output;
run;

data final4b;
merge final4 (in=a) final4a (in=b);
by odd;
run;

data final4c;
set final4b;

if sort3 ne 0 and flg_n3='Y' then n3='';
if sort3 ne 0 and flg_n4='Y' then n4='';
if sort3 ne 0 and flg_n5='Y' then n5='';
if sort3 ne 0 and flg_n96='Y' then n96='';
run;

data tflds.T_15_02_01_06 (keep=mhbodsys mhdecod trt:);
set final4c (keep=mhbodsys mhdecod n3 n4 n5 n96 n99 e3 e4 e5 e96 e99 p3 p4 p5 p96 p99);
length trt_4 $20. trt_5 $20. trt_3 $20. trt_96 $20. trt_99 $20. ;

if compress(n3) ne '0' then trt_3=compress(n3)||compress(p3)||compress(e3);
else if compress(n3)='0' then trt_3=compress(n3);

if compress(n4) ne '0' then trt_4=compress(n4)||compress(p4)||compress(e4);
else if compress(n4)='0' then trt_4=compress(n4);

if compress(n5) ne '0' then trt_5=compress(n5)||compress(p5)||compress(e5);
else if compress(n5)='0' then trt_5=compress(n5);

if compress(n96) ne '0' then trt_96=compress(n96)||compress(p96)||compress(e96);
else if compress(n96) eq '0' then trt_96=compress(n96);

if compress(n99) ne '0' then trt_99=compress(n99)||compress(p99)||compress(e99);
else if compress(n99)='0' then trt_99=compress(n99);

run;

data paging;
set final4c;
by sort2 odd odd2;
cnt+1;
page=ceil(cnt/10);
call symput("page",compress(put(page,best.)));

run;

options nonumber 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=70, halfblnk=N);

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

ods path stdlib.t106343 (read) ;
ods results off;
ods rtf toc_data/* contents*/ file="/cvn/projects/prj/data/000000106343/TFL/dev/Tables/&tfino..rtf" style=t106343 startpage=yes head
ery=1440 footery=1440 ;
ods noproctitle;
%do i=1 %to &page;

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

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

    if missing(column) then call symput('NOOBS','1');

    /* Amend title as needed */
    _firtitl="Table 15.2.1.6 Summary of Medical History - Safety Population";
    _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.)));
    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 spacing=1 style={cellspacing=2pt } missing nowd split = '$' %if &i=1 %then %do; contents=
' ' %end; %else %do; contents='' %end;;
    column flag page sort2 odd odd2 ("System Organ Class" (|~ {Preferred Term}" column)) ("THSm2.2 $(N=&trt4) &linebot" ("n (%) Ev
ents" n4 p4 e4)) ("mCC$(N=&trt5) &linebot" ("n (%) Events" n5 p5 e5))
        ("SA $(N=&trt3) &linebot" ("n (%) Events" n3 p3 e3)) ("Product Test $(N=&trt96) &linebot" (" n (%) Events" n96 p96
e96))
        ("Overall$Safety$(N=&trt99) &linebot" ("n (%) Events" n99 p99 e99)); ;
    define flag      / order order = internal noprint;
    define page      / order order = internal noprint;
    define sort2     / order order=internal noprint;
        define odd      / order order=internal noprint;
        define odd2     / order order=internal noprint;

    define column    / group style={just=left cellwidth=4.2cm} style(header)={just=right} "";
    define n3        / display style={just=c cellwidth=0.5cm} style(header)={just=right} "" ;
    define p3        / display style={just=d cellwidth=0.8cm} style(header)={just=l} "";
    define e3        / display style={JUST=center cellwidth=0.8cm} style(header)={just=left} "";
    define n4        / display style={just=d cellwidth=0.5cm} style(header)={just=right} "";
    define p4        / display style={just=d cellwidth=0.8cm} style(header)={just=left} "";
    define e4        / display style={JUST=center cellwidth=0.8cm } style(header)={just=l} "";
    define n5        / display style={just=d cellwidth=0.5cm} style(header)={just=left} "";
    define p5        / display style={just=d cellwidth=0.8cm} style(header)={just=l} "";
    define e5        / display style={JUST=center cellwidth=0.8cm} style(header)={just=l} "";
    define n96       / display style={just=d cellwidth=0.5cm} style(header)={just=right} "";
    define p96       / display style={just=d cellwidth=0.8cm} style(header)={just=l} "";
    define e96       / display style={JUST=center cellwidth=0.8cm} style(header)={just=l} "";
    define n99       / display style={just=d cellwidth=0.5cm} style(header)={just=right} "";
    define p99       / display style={just=d cellwidth=0.8cm} style(header)={just=left} "";
    define e99       / display style={JUST=center cellwidth=0.8cm} style(header)={just=l} "";

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

```



```

break after page / page;

%if &noobs ne 1 %then %do;
compute after odd;
line " ";
endcomp;
%end;

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

compute after page/style={just=center cellwidth=5cm protectspecialchars=off};
%if &NOOBS.=1 %then %do;
line "No medical histories were recorded";
line " ";
%end;
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: "Product Test" refers to all subjects who tested the THS product but were not randomized.';
line 'Note: The "Overall Safety" refers to all subjects in the Safety Population.';
line 'Note: mCC = Menthol conventional cigarettes; SA = Smoking abstinence; THSm2.2 = Tobacco Heating System 2.2 Menthol.';
line 'Note: Percentages are based on the number of subjects indicated in the column header (N).';
line ' ';
line 'Appendix 15.3.1.9.';
line "Study ID: ZRHM-REXA-08-US          Program: &TFLprg          Status: &status" &_blankn.*"\-\" "&sysdate" &_blankn.*"\-\" "(P
age &i of &page)";
endcomp;

run;
%end;
ods rtf close;
ods results on;
ods path sashelp.tmplmst (read);

%mend ;

%outrtf(blankn=30, halfblnk=N);

%m_logchk;

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

```