

```

/*****
** STUDY ID : 000000106343
**
** PROGRAM NAME : t_pe.sas
**
** DATE : 28May2015
**
** PROGRAMMER : cvn_aramasah
**
** PURPOSE : QC the table Summary of physical examination of body systems - safety population (t_15_2_6.23)
**
** INPUT DATA : ADAM.ADSL, ADAM.ADPE
**
** OUTPUT DATA :
**
** SAS MACROS USED :
**
** MODIFICATIONS : DATE : MODIFIED BY : NOTES :
**
**-----**
** PROGRAMMED USING SAS VERSION 9.3 **
** COPYRIGHT (C) 2015 BY COVANCE, PRINCETON NJ - USA - ALL RIGHTS RESERVED **
**-----**/
options notes source source2 nofullstimer validvarname=upcase missing=' ';
ods _all_ close;
ods listing;
%m_printto;
options notes nosource replace;
proc datasets lib=work nolist memtype=data kill; quit;

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

%let tflno=T_15_02_06_23;
%let TFLprg=t_pe.sas;
%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 numbers for column headers*/
data adsl;
    set adam.adsl;
    if trt01a='THSm2.2' then trt01an=1;
    if trt01a='mCC' then trt01an=2;
    if trt01a='SA' then trt01an=3;
    where safaf1 = '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='THSm2.2';
    output;
    trt01an=2;

```

```

trt01a='mCC';
output;
trt01an=3;
trt01a='SA';
output;
trt01an=99;
trt01a='Overall Safety';
output;
run;

data tot2a;
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(put(count,best.)));
run;

data tot2;
set tot2a;
trta=trt01a;
trtan=trt01an;
drop trt01an trt01a;
run;

/*Bring in appropriate data from ADPE*/

proc sort data=adam.adpe out=adpe (where=( safaf1='Y' and avalc ne 'Not Examined' and paramn in (2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12,
13, 14, 15)));
by paramn avisitn;
run;

data adpe;
set adpe;
if ablfl='Y' then do;
avisitn=100;
avisit='Baseline';
end;

if avisitn ge 100;
/*for all visits. for the counts of Normal, Abnormal NCS, Abnormal CS at baseline. Normal to Normal, Abnormal NCS to Normal, Abnorma
l CS to Normal like counts for all other visits */
if ablfl='Y' then do;
if PECLSIG ne '' then cat2=strip(avalc)||', '||strip(PECLSIG);
else cat2=avalc;
end;
if shift1 ne '' then cat2=shift1;
/*for other than baseline visits. for the counts of Normal, Abnormal NCS, Abnormal CS */
if PECLSIG ne '' then cat1=strip(avalc)||' '||strip(PECLSIG);
else cat1=avalc;

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

run;

data baseline;
set adpe;
if avisitn=100;
run;

data other_visits;
set adpe;
if avisitn gt 100;
run;

/* for baseline counts */
proc freq data=baseline noprint;
table paramn*paramcd*param*trta*avisitn*avisit*avalc*cat2/ out=baseline_freq;
run;

proc sort data=baseline_freq;
by paramn paramcd param avisitn avisit avalc cat2;
run;

```

```

proc transpose data=baseline_freq out=baseline_trans (drop=_name_ _label_);
by paramn paramcd param avisitn avisit avalc cat2;
id trta;
var count;
run;

/* for all other visits, sub group-1 */
proc freq data=other_visits noprint;
table paramn*paramcd*param*trta*avisitn*avisit*avalc*cat1/ out=other_visits_freq_1;
run;

proc sort data=other_visits_freq_1;
by paramn paramcd param avisitn avisit avalc cat1;
run;

proc transpose data=other_visits_freq_1 out=other_visits_trans_1 (drop=_name_ _label_);
by paramn paramcd param avisitn avisit avalc cat1;
id trta;
var count;
run;

/* for all other visits, sub group-2 */
proc freq data=other_visits noprint;
table paramn*paramcd*param*trta*avisitn*avisit*avalc*cat2/ out=other_visits_freq_2;
run;

proc sort data=other_visits_freq_2;
by paramn paramcd param avisitn avisit avalc cat2;
run;

proc transpose data=other_visits_freq_2 out=other_visits_trans_2 (drop=_name_ _label_);
by paramn paramcd param avisitn avisit avalc cat2;
id trta;
var count;
run;

proc sort data=other_visits_trans_2;
by paramn paramcd param avisitn avisit avalc cat2;
proc sort data=other_visits_trans_1;
by paramn paramcd param avisitn avisit avalc cat1;
proc sort data=baseline_trans;
by paramn paramcd param avisitn avisit avalc cat2;
run;
/* putting all numbers together */
data all_visits_counts;
length stat $200.;
set baseline_trans other_visits_trans_1 other_visits_trans_2;
by paramn paramcd param avisitn avisit avalc;
stat=cat1;
if cat2 ne '' then stat=cat2;
run;

data order;
set all_visits_counts;
/* sort */
if cat2 ne '' then do;
if upcase(cat2)='NORMAL' then sort=1;
else if upcase(cat2)='ABNORMAL, NCS' then sort=2;
else if upcase(cat2)='ABNORMAL, CS' then sort=3;
end;
if stat ne '' and sort = . then do;
if (stat)= 'Normal' then sort=4;
if (stat)= 'Normal to Normal' then sort=5;
if (stat)= 'Abnormal, NCS to Normal' then sort=6;
if (stat)= 'Abnormal, CS to Normal' then sort=7;
if (stat)= 'Abnormal NCS' then sort=8;
if (stat)= 'Normal to Abnormal, NCS' then sort=9;
if (stat)= 'Abnormal, NCS to Abnormal, NCS' then sort=10;
if (stat)= 'Abnormal, CS to Abnormal, NCS' then sort=11;
if (stat)= 'Abnormal CS' then sort=12;
if (stat)= 'Normal to Abnormal, CS' then sort=13;
if (stat)= 'Abnormal, NCS to Abnormal, CS' then sort=14;
if (stat)= 'Abnormal, CS to Abnormal, CS' then sort=15;
end;

run;

proc sort data=order;

```

```

by paramn avisitn sort;
run;

data dummy;
length stat $204.;
stat='Normal' ; sort=1;avisitn=100;output;
stat='Abnormal, NCS' ; sort=2;avisitn=100;output;
stat='Abnormal, CS' ; sort=3;avisitn=100;output;

stat= 'Normal';sort=4;avisitn=106;output;
stat= 'Normal';sort=4;avisitn=130;output;
stat= 'Normal';sort=4;avisitn=160;output;
stat= 'Normal';sort=4;avisitn=191;output;

stat= 'Normal to Normal';sort=5;avisitn=106;output;
stat= 'Normal to Normal';sort=5;avisitn=130;output;
stat= 'Normal to Normal';sort=5;avisitn=160;output;
stat= 'Normal to Normal';sort=5;avisitn=191;output;

stat= 'Abnormal, NCS to Normal';sort=6;avisitn=106;output;
stat= 'Abnormal, NCS to Normal';sort=6;avisitn=130;output;
stat= 'Abnormal, NCS to Normal';sort=6;avisitn=160;output;
stat= 'Abnormal, NCS to Normal';sort=6;avisitn=191;output;

stat= 'Abnormal, CS to Normal'; sort=7;avisitn=106;output;
stat= 'Abnormal, CS to Normal'; sort=7;avisitn=130;output;
stat= 'Abnormal, CS to Normal'; sort=7;avisitn=160;output;
stat= 'Abnormal, CS to Normal'; sort=7;avisitn=191;output;

stat= 'Abnormal NCS'; sort=8;avisitn=106;output;
stat= 'Abnormal NCS'; sort=8;avisitn=130;output;
stat= 'Abnormal NCS'; sort=8;avisitn=160;output;
stat= 'Abnormal NCS'; sort=8;avisitn=191;output;

stat= 'Normal to Abnormal, NCS' ; sort=9;avisitn=106;output;
stat= 'Normal to Abnormal, NCS' ; sort=9;avisitn=130;output;
stat= 'Normal to Abnormal, NCS' ; sort=9;avisitn=160;output;
stat= 'Normal to Abnormal, NCS' ; sort=9;avisitn=191;output;

stat= 'Abnormal, NCS to Abnormal, NCS' ; sort=10;avisitn=106;output;
stat= 'Abnormal, NCS to Abnormal, NCS' ; sort=10;avisitn=130;output;
stat= 'Abnormal, NCS to Abnormal, NCS' ; sort=10;avisitn=160;output;
stat= 'Abnormal, NCS to Abnormal, NCS' ; sort=10;avisitn=191;output;

stat= 'Abnormal, CS to Abnormal, NCS' ; sort=11;avisitn=106;output;
stat= 'Abnormal, CS to Abnormal, NCS' ; sort=11;avisitn=130;output;
stat= 'Abnormal, CS to Abnormal, NCS' ; sort=11;avisitn=160;output;
stat= 'Abnormal, CS to Abnormal, NCS' ; sort=11;avisitn=191;output;

stat= 'Abnormal CS' ; sort=12;avisitn=106;output;
stat= 'Abnormal CS' ; sort=12;avisitn=130;output;
stat= 'Abnormal CS' ; sort=12;avisitn=160;output;
stat= 'Abnormal CS' ; sort=12;avisitn=191;output;

stat= 'Normal to Abnormal, CS' ; sort=13;avisitn=106;output;
stat= 'Normal to Abnormal, CS' ; sort=13;avisitn=130;output;
stat= 'Normal to Abnormal, CS' ; sort=13;avisitn=160;output;
stat= 'Normal to Abnormal, CS' ; sort=13;avisitn=191;output;

stat= 'Abnormal, NCS to Abnormal, CS' ; sort=14;avisitn=106;output;
stat= 'Abnormal, NCS to Abnormal, CS' ; sort=14;avisitn=130;output;
stat= 'Abnormal, NCS to Abnormal, CS' ; sort=14;avisitn=160;output;
stat= 'Abnormal, NCS to Abnormal, CS' ; sort=14;avisitn=191;output;

stat= 'Abnormal, CS to Abnormal, CS' ; sort=15;avisitn=106;output;
stat= 'Abnormal, CS to Abnormal, CS' ; sort=15;avisitn=130;output;
stat= 'Abnormal, CS to Abnormal, CS' ; sort=15;avisitn=160;output;
stat= 'Abnormal, CS to Abnormal, CS' ; sort=15;avisitn=191;output;

run;

data dummy2;
set dummy;
paramn=2;output;
paramn=3;output;
paramn=4;output;
paramn=5;output;
paramn=6;output;

```

```

paramn=7;output;
paramn=8;output;
paramn=9;output;
paramn=10;output;
paramn=11;output;
paramn=12;output;
paramn=13;output;
paramn=14;output;
paramn=15;output;
run;

proc sort data=dummy2;
by paramn avisitn sort stat;
proc sort data=order;
by paramn avisitn sort stat;
run;

data testing;
merge order(in=order drop=stat) dummy2;
by paramn avisitn sort;
run;

proc sort data=testing (drop=avalc cat1 cat2);
by paramn avisitn sort;
run;

data final_1;
set testing;
if paramn=2 then paramcd='HEENT';
if paramn=3 then paramcd='THYROID';
if paramn=4 then paramcd='HEART';
if paramn=5 then paramcd='CHEST';
if paramn=6 then paramcd='LUNGS';
if paramn=7 then paramcd='GASTRO';
if paramn=8 then paramcd='CVS';
if paramn=9 then paramcd='NEURO';
if paramn=10 then paramcd='SKIN';
if paramn=11 then paramcd='BACK';
if paramn=12 then paramcd='MUSCULO';
if paramn=13 then paramcd='ABDOMEN';
if paramn=14 then paramcd='DENTN';
/*if paramn=15 then paramcd='HEENT';*/
if avisitn=100 then avisit='Baseline';
if avisitn=106 then avisit='Day 6/Discharge Confinement';
if avisitn=130 then avisit='Day 30';
if avisitn=160 then avisit='Day 60';
if avisitn=191 then avisit='Day 91/Discharge Ambulatory';

run;

data final_2;
set final_1;
if paramcd ne '';
run;
/* percentages */

data final_3;
set final_2;

if thsm2_2=&trt1. and thsm2_2 ne . then ths=strip(put(THSM2_2,best.))||' (100)';
else if THSM2_2 ne . and thsm2_2 ne &trt1. then ths=strip(put(THSM2_2,best.))||' ('|| strip(put((THSM2_2/&trt1.)*100,5.1))||')';
else ths='0';

if mcc=&trt2. and mcc ne . then cc=strip(put(mcc,best.))||' (100)';
else if mcc ne . and mcc ne &trt2. then cc=strip(put(mcc,best.))||' ('|| strip(put((mcc/&trt2.)*100,5.1))||')';
else cc='0';

if sa=&trt3. and sa ne . then sa_=strip(put(sa,best.))||' (100)';
else if sa ne . and sa ne &trt3. then sa_=strip(put(sa,best.))||' ('|| strip(put((sa/&trt3.)*100,5.1))||')';
else sa_='0';

if overall_safety=&trt99. and overall_safety ne . then overall=strip(put(overall_safety,best.))||' (100)';
else if overall_safety ne . and overall_safety ne &trt99. then overall=strip(put(overall_safety,best.))||' ('|| strip(put((overall_s
afety/&trt99.)*100,5.1))||')';
else overall='0';

drop THSM2_2 mcc sa overall_safety;

```

```

if OVERALL='0' then delete;
if indexw(upcase(avisit),'UNSCHEDULED') ne 0 then delete;
run;

data final_3;
  set final_3;
  by paramn paramcd avisitn avisit;
/*   if missing(ov_sub) and ln>8 then ln=1; */
/*   else ln+1;*/
/*   if ln=1 then page+1;*/
/*   call symput("page",compress(put(page,best.)));*/
/* flag=1;*/

      if first.avisitn and ln ge 6 then ln=1; /*Amend to look presentable, and avoid page overflows*/
      else ln+1;
      if ln=1 then page+1;
      call symput("page",compress(put(page,best.)));
flag=1;
run;

/*proc sort data=final_3 out=page(keep=paramn paramcd avisitn avisit) nodupkey;*/
/*by paramn paramcd avisitn avisit;*/
/*run;*/
/**/
/*data page;*/
/*set page;*/
/*page_num=_n_;*/
/*run;*/
/**/
/*proc sort data=page;*/
/*by paramn paramcd avisitn avisit;*/
/*proc sort data=final_3;*/
/*by paramn paramcd avisitn avisit;*/
/*run;*/

data final_4;
merge final_3* page;
by paramn paramcd avisitn avisit;

if stat='Normal' then stat='Normal - n (%)';
if stat in ('Abnormal, NCS' 'Abnormal NCS') then stat='Abnormal NCS - n (%)';
if stat in ('Abnormal, CS' 'Abnormal CS') then stat='Abnormal CS - n (%)';

if stat='Normal to Normal' then stat='Normal to Normal - n (%)';
if stat='Abnormal, NCS to Normal' then stat='Abnormal NCS to Normal - n (%)';
if stat='Abnormal, CS to Normal' then stat='Abnormal CS to Normal - n (%)';

if stat='Normal to Abnormal, NCS' then stat='Normal to Abnormal NCS - n (%)';
if stat='Abnormal, NCS to Abnormal, NCS' then stat='Abnormal NCS to Abnormal NCS - n (%)';
if stat='Abnormal, CS to Abnormal, NCS' then stat='Abnormal CS to Abnormal NCS - n (%)';

if stat='Normal to Abnormal, CS' then stat='Normal to Abnormal CS - n (%)';
if stat='Abnormal, NCS to Abnormal, CS' then stat='Abnormal NCS to Abnormal CS - n (%)';
if stat='Abnormal, CS to Abnormal, CS' then stat='Abnormal CS to Abnormal CS - n (%)';

run;

proc sort data=final_4;
by page paramn paramcd avisitn avisit sort;
run;

data final_5;
set final_4;
by page paramn paramcd avisitn avisit sort;
/*if first.page_num then do;*/
/*paramcd=paramcd;*/
/*avisit=avisit;*/
/*end;*/
/*else do;*/
/*paramcd='';*/
/*avisit='';*/
/*end;*/
avisit=avisit;
if sort not in (1, 2, 3, 4, 8, 12) then stat="      ||strip(stat);
else stat=strip(stat);
/*   call symput("page",compress(put(page_num,best.)));*/
/* flag=1;*/

```

```

run;

proc sort data=final_5 out=lastpage;
by paramn;
run;

data lastpage_1;
set lastpage;
by paramn;
if last.paramn ;
run;

proc sort data=lastpage_1;
by avisitn paramn;
run;

data lastpage_2;
set lastpage_1;
by avisitn paramn;
if last.avisitn then call symput('maxpage', trim(left(put(page,best.))));
run;

proc sql noprint;

create table tflds.&tflno as
select paramcd, paramn, param, avisitn, avisit, sort, stat, ths , cc as mcc , sa_ as sa, overall as overall_safety, page
from final_5
order by page, paramn, avisitn, sort;

quit;
/*proc sort data=final_4;*/
/*by page_num paramn paramcd avisitn avisit sort;*/
/*run;*/
/*data final_5;*/
/*set final_4;*/
/*by page_num paramn paramcd avisitn avisit sort;*/
/*if first.page_num then do;*/
/*paramcd=paramcd;*/
/*avisit=avisit;*/
/*end;*/
/*else do;*/
/*paramcd='';*/
/*avisit='';*/
/*end;*/
/*avisit=avisit;*/
/*if sort not in (1, 2, 3, 4, 8, 12) then stat="      ||strip(stat);*/
/*else stat=strip(stat);*/
/*      call symput("page",compress(put(page_num,best.)));*/
/*  flag=1;*/
/**/
/*run;*/
proc sort data=final_5;
by flag paramn avisitn sort;
run;

options number nodate orientation=landscape /*papersize=&p_gsize*/ 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.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;
%let supfl=0;

```

```

%let npage=%eval(&i);

data comp;
    set final_5 end=eof;
    where page=&i;
    call symput('paramn',STRIP(PUT(paramn,BEST.)));

    /* Amend title as needed */
    _firtitl="Table 15.2.6.23 Summary of Physical Examination of Body Systems - Safety Population";
    _SECONDTL="Safety Time Period: Randomized Period";
    _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('_SECONDTL', strip(_SECONDTL));
        call symput('_blankn', compress(put(len,best.)));
    end;
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 = '$';;
    column flag page PARAMn param avisitn avisit sort stat
        ("THSm2.2 $(N=&trt1) &linebot" ths)
        ("mCC $(N=&trt2) &linebot" cc)
        ("SA $(N=&trt3) &linebot" sa_)
        ("Overall Safety$(N=&trt99) &linebot" overall);

    define flag          / order order=internal noprint;
    define page          / order order=internal noprint;
    define paramn        / order order=internal noprint;
    define avisitn       / order order = internal noprint;
    define sort          / order order=internal noprint;
    define param          /order style={just=1 cellwidth=0.4 cm} style(header)={just=left} 'Body$System';
    define avisit        / order style={just=1 cellwidth=0.5 cm} style(header)={just=left} 'Study$Day';
    define stat          / display style={indent=10 cellwidth=1 cm} style(header)={just=left}'Statistic'; /* 11) JMH 16Sep2014 */
    define ths           / display style={just=left cellwidth=0.2 cm} style(header)={just=left} '';
    define cc            / display style={just=left cellwidth=0.2 cm} style(header)={just=left} '';
    define sa_           / display style={just=left cellwidth=0.2 cm} style(header)={just=left} '';
    define overall       / display style={just=left cellwidth=0.2 cm} style(header)={just=left} '';

compute stat;
if substr(stat,1,1)=' ' then do;
call define (_col_,'style','style={backgroundcolor=white indent=50}');
end;
endcomp;

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

break after page / page;

compute after avisitn;*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." ;
line "&linebot";
line "\b\fs24\sa24&_SECONDTL." ;
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; NCS = no
t clinically significant; CS = Clinically Significant.";
line "Note: Percentages are based on the number of subjects indicated in the column header (N).";
line ' ';

```



```
line 'Appendix 15.3.6.12';
line "Study ID:ZRHM-REXA-08-US   Program:&TFLprg   Status: &status" &_blankn.*"\~\~" "&sysdate" &_blankn.*"\~\~" "(Page &i of &pag
e)";
endcomp;
run;
%end;
ods rtf close;
ods results on;
ods path sashelp.tmplmst (read);

%mend ;

%outrtf(blankn=40, halfblnk=N);

ods listing;

ods listing close;

/*proc datasets lib=work nolist memtype=data kill; quit;*/

%m_logchk;
```