

```

proc datasets lib=work nolist memtype=data kill; quit;
/*=====
*Covance Study ID      : 000000106331
*Program Name          : t_protviol.sas
*Purpose               : Table 15.2.1.3.1 Summary of Protocol Deviations â€œ Safety Population
*Input Data            : adam.adsl, adam.addv
*Output Data           : tflds.T_15_02_01_03_01
*Macros Called         : %m_printto, %trans, %outtrtf, %m_logchk2
*Programmed by        : Ranju Gautam
*Creation Date         : 2015-06-09
*=====
*Modification History
*Date      Initials   No. Reason;
*=====*/

%m_printto(route=YES);

data adsl;
set adam.adsl;
where ( SAFBFL='Y' or SAFAFL = 'Y') and trt01an in (3 4 5);
run;

data adsl0;
set adsl adsl(in=a);
if a then trt01an=99;
run;

proc sql noprint;

*for block1a;
select count (distinct usubjid)  into :ths from adsl0 where trt01an=4 ;
select count (distinct usubjid)  into :mc from adsl0 where trt01an=5 ;
select count (distinct usubjid)  into :sa from adsl0 where trt01an=3 ;
select count (distinct usubjid)  into :tot from adsl0 where trt01an=99;

quit;

data addv;
set adam.addv;
where (SAFBFL='Y' or SAFAFL = 'Y' ) and trtan in (3,4,5);
run;

data dv;
set addv addv(in=a);
if a then trtan=99;
run;

proc sort data=dv;
by usubjid;
run;

proc sql noprint;

create table block1 as select TRTAN, count(distinct usubjid ) as count  from dv
group by TRTAN ;

create table block2 as select TRTAN, count(distinct usubjid ) as count  from dv where
upcase(DVSIG)='MAJOR' group by TRTAN;

create table block3 as select TRTAN, count(distinct usubjid ) as count  from dv where
upcase(DVSIG)!='MINOR' AND upcase(EVALCAT)='NON EVALUABLE' group by TRTAN ;

create table block4 as select TRTAN, EVALCAT, dvcat, count(distinct usubjid ) as count,
2 as ord from dv where upcase(DVSIG)!='MINOR' AND upcase(EVALCAT)='NON EVALUABLE'
group by TRTAN, EVALCAT, dvcat order by ord, evalcat, dvcat;

*without evaluable impact;
create table block5 as select TRTAN, count(distinct usubjid ) as count  from dv where
upcase(DVSIG)!='MINOR' AND upcase(EVALCAT)='EVALUABLE' group by TRTAN;

*without evaluable impact with param;
create table block6 as select TRTAN, EVALCAT, dvcat, count(distinct usubjid ) as count  from dv where

```

```
upcase(DVSIG)!='MINOR' AND upcase(EVALCAT)='EVALUABLE' group by TRTAN , evalcat, dvcat order by evalcat, dvcat;
```

```
*Minor;
```

```
create table block7 as select TRTAN, DVSIG, count(distinct usubjid ) as count from dv where  
upcase(DVSIG)='MINOR' and dvcat ne '' group by TRTAN ,DVSIG order by DVSIG ;
```

```
*Minor with param;
```

```
create table block8 as select TRTAN, DVSIG, dvcat, count(distinct usubjid ) as count from dv where  
upcase(DVSIG)='MINOR' and dvcat ne '' group by TRTAN ,DVSIG,dvcat order by DVSIG,dvcat ;
```

```
**for events;
```

```
create table block1a as select TRTAN, count(usubjid ) as count from dv  
group by TRTAN ;
```

```
create table block2a as select TRTAN, count( usubjid ) as count from dv where  
upcase(DVSIG)!='MINOR' group by TRTAN;
```

```
create table block3a as select TRTAN, count( usubjid ) as count from dv where  
upcase(DVSIG)!='MINOR' AND upcase(EVALCAT)='NON EVALUABLE' group by TRTAN ;
```

```
create table block4a as select TRTAN, EVALCAT, dvcat , count( usubjid ) as count,  
2 as ord from dv where upcase(DVSIG)!='MINOR' AND upcase(EVALCAT)='NON EVALUABLE'  
group by TRTAN, EVALCAT, dvcat order by ord, evalcat, dvcat;
```

```
*without evaluable impact;
```

```
create table block5a as select TRTAN, count( usubjid ) as count from dv where  
upcase(DVSIG)!='MINOR' AND upcase(EVALCAT)='EVALUABLE' group by TRTAN ;
```

```
*without evaluable impact with param;
```

```
create table block6a as select TRTAN, EVALCAT, dvcat, count( usubjid ) as count from dv where  
upcase(DVSIG)!='MINOR' AND upcase(EVALCAT)='EVALUABLE' group by TRTAN , EVALCAT,dvcat order by evalcat, dvcat;
```

```
*Minor;
```

```
create table block7a as select TRTAN, DVSIG, count( usubjid ) as count from dv where  
upcase(DVSIG)='MINOR' and dvcat ne '' group by TRTAN ,DVSIG order by DVSIG ;
```

```
*Minor with param;
```

```
create table block8a as select TRTAN, DVSIG, DVCAT , count( usubjid ) as count from dv where  
upcase(DVSIG)='MINOR' and dvcat ne '' group by TRTAN ,DVSIG, dvcat order by DVSIG,dvcat ;
```

```
quit;
```

```
%macro trans(inds=,outds=t,var=);
```

```
proc transpose data=&inds out=&&outds.&inds(drop=_name_) prefix=_;
```

```
var count;
```

```
id trtan ;
```

```
by &var ;
```

```
run;
```

```
%mend trans;
```

```
%trans(inds=block1,var=);
```

```
%trans(inds=block2,var=);
```

```
%trans(inds=block3,var=);
```

```
%trans(inds=block4,var=evalcat dvcat);
```

```
%trans(inds=block55,var=);
```

```
%trans(inds=block6,var=evalcat dvcat);
```

```
%trans(inds=block7,var=DVSIG);
```

```
%trans(inds=block8,var=DVSIG dvcat );
```

```
%trans(inds=block1a,var=);
```

```
%trans(inds=block2a,var=);
```

```
%trans(inds=block3a,var=);
```

```
%trans(inds=block4a,var=evalcat dvcat);
```

```
%trans(inds=block55a,var=);
```

```
%trans(inds=block6a,var=evalcat dvcat);
```

```
%trans(inds=block7a,var=DVSIG);
```

```
%trans(inds=block8a,var=DVSIG dvcat );
```

```
data fin;
```

```
set tblock1(in=_11 ) tblock2(in=_22) tblock3(in=_33) tblock4(in=_44) tblock55(in=_55)
```

```
tblock6(in=_66) tblock7(in=_77) tblock8(in=_88);
```

```

if _11 then ord=1;
if _22 then ord=2;
if _33 then ord=3;
if _44 then ord=4;
if _55 then ord=6;
if _66 then ord=7;
if _77 then ord=8;
if _88 then ord=9;

run;

data fina;
set tblock1a(in=_11 ) tblock2a(in=_22) tblock3a(in=_33) tblock4a(in=_44) tblock55a(in=_55)
    tblock6a(in=_66) tblock7a(in=_77) tblock8a(in=_88);
if _11 then ord=1;
if _22 then ord=2;
if _33 then ord=3;
if _44 then ord=4;
if _55 then ord=6;
if _66 then ord=7;
if _77 then ord=8;
if _88 then ord=9;

run;

proc sort data= fin;
    by ord dvcat ;
run;

proc sort data= fina;
    by ord dvcat ;
run;

data fin1;
    length c1-c4 c1_ c2_ c3_ c4_ $ 30 term $ 200 dvcat1 $ 60;
    merge fin (in=a) fina(in=b rename=(_3=_3e _4=_4e _5=_5e _99=_99e) drop=evalcat DVSIG );
    by ord dvcat ;

if _4 ne . then c1=strip(put(_4,best.))||' (||strip(put(_4*100/&ths,5.1))||)' ;
if _5 ne . then c2=strip(put(_5,best.))||' (||strip(put(_5*100/&mc ,5.1))||)' ;
if _3 ne . then c3=strip(put(_3,best.))||' (||strip(put(_3*100/&sa,5.1))||)' ;
if _99 ne . then c4=strip(put(_99,best.))||' (||strip(put(_99*100/&tot,5.1))||)' ;

*splitting n (%) and evests separately;
if _4 ne . then c1_=strip(put(_4e, best.));
if _5 ne . then c2_=strip(put(_5e, best.));
if _3 ne . then c3_=strip(put(_3e, best.));
if _99 ne . then c4_=strip(put(_99e, best.));

**making all text for dvcat;
if dvcat='ASSESSMENT MISSING' then dvcat1='Assessment missing';
if dvcat='INFORMATION ENTRY IN THE INCORRECT LOCATION' then dvcat1='Information entry in the incorrect location';
if dvcat='MEASURE REPEATED' then dvcat1='Measure repeated';
if dvcat='OTHER' then dvcat1='Other';
if dvcat='PART OF THE SPIROMETRY PROCEDURE RECORDINGS INVERTED' then dvcat1='Part of the spirometry procedure recordings inverted';
if dvcat='REPEATED MEASURE DONE IN ERROR' then dvcat1='Repeated measure done in error';
if dvcat='TIME DEVIATION' then dvcat1='Time deviation';
if dvcat='TIME OF VITAL SIGNS ASSESSMENT MISSING' then dvcat1='Time of vital signs assessment missing';
if dvcat='VITAL SIGNS DATA UNAVAILABLE IN SOURCE DATA' then dvcat1='Vital signs data unavailable in source data';
if dvcat='CONCOMITANT MEDICATION' then dvcat1='Concomitant medication';
if dvcat='VIOLATION' then dvcat1='Violation';
if dvcat='TIME MISSING' then dvcat1='Time missing';
if dvcat='MIS-USE OF PRODUCT IN PERIOD 1' then do; dvcat1='Mis-use of product in period 1'; subord=91; end;
if dvcat='MIS-USE OF PRODUCT IN PERIOD 2' then do; dvcat1='Mis-use of product in period 2'; subord=92; end;
if dvcat='MIS-USE OF PRODUCT IN PERIOD 3' then do; dvcat1='Mis-use of product in period 3'; subord=93; end;
if dvcat='MIS-USE OF PRODUCT IN PERIOD 4' then do; dvcat1='Mis-use of product in period 4'; subord=94; end;

*other should come as the last dvcat;
if ord in (4 7 9) and compress(dvcat1)='Other' then subord=2;

if ord=1 then term='Total';
if ord=2 then term='Major';
if ord=3 then term='With evaluability impact';
if ord=4 then term=' '||strip(dvcat1);

```

```

if ord=6 then term='Without evaluability impact';
if ord=7 then term='    '||strip(dvcat1);

if ord=8 then term='Minor';

if ord=9 then term='    '||strip(dvcat1);
if evalcat='NON EVALUABLE' and dvcat='MIS-RANDOMIZATION' then term='    Mis-Randomization';
run;

proc sort data=fin1;
  by ord subord ;
run;

%let tflno=T_15_02_01_03_01;

data tflds.T_15_02_01_03_01;
  set fin1 ;
run;

data paging;
  set fin1;
  by ord subord ;
  cnt+1;

page = ceil(cnt/12);

  call symput("page",compress(put(page,best.)));
  if ord in (1 2 3 4) then brk=1;
  if ord in (5 6 7) then brk=2;
  if ord in (8 9) then brk=3;
run;

data final_page;
  set paging;
run;

*****;
*create output report ;
*****;

options number nodate orientation=landscape missing=' ';
ods escapechar='$';
%let linetop = \brdrt\brdrs\brdrw30;
%let linebot = \brdrb\brdrs\brdrw30;

%macro outrtf(blankn=130, halfblnk=N, dsn=);

  %let title1 = %nrstr(Table 15.2.1.3.1 Summary of Protocol Deviations - Safety Population);

  %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;

%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/&TFL_Part./Tables/&tflno..rtf" style=t106343 startpage=yes headery=1
440 footery=1440 ;
ods noproctitle;

%do i=1 %to &page;
  title ;
  footnote;
  %let wd=0;

```

```

ods proclabel = ' ';

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

    _firtitl="%title1.";
    _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 listing close;

proc report data = comp headline headskip nowd split = '$' %if &i=1 %then %do; contents=' ' %end; %else %do; contents='' %end;;;
column page brk cnt term
    ("THSm2.2$(N=%sysfunc(compress(&ths)))" c1 c1_)
    ("mCC$(N=%sysfunc(compress(&mc)))" c2 c2_)
    ("SA$(N=%sysfunc(compress(&sa)))" c3 c3_)
    ("Overall Safety$(N=%sysfunc(compress(&tot)))" c4 c4_);

define page          / noprint order ;
define cnt           / order order = internal noprint;
define brk           / order order = internal noprint;
define term           /"Classification of Deviation" group order=internal style={just=left cellwidth=1.6cm asis=on} style(header)={
just=left} flow;

define c1             /"n (%)" display style={just=c cellwidth=0.3cm} style(header)={just=left} ;
define c1_            /"Events" display style={just=c cellwidth=0.3cm} style(header)={just=left} ;

define c2             /"n (%)" display style={just=c cellwidth=0.4cm} style(header)={just=center} ;
define c2_            /"Events" display style={just=c cellwidth=0.3cm} style(header)={just=center} ;

define c3             /"n (%)" display style={just=c cellwidth=0.4cm} style(header)={just=center} ;
define c3_            /"Events" display style={just=c cellwidth=0.3cm} style(header)={just=center} ;

define c4             /"n (%)" display style={just=c cellwidth=0.4cm} style(header)={just=center} ;
define c4_            /"Events" display style={just=c cellwidth=0.2cm} style(header)={just=center} ;

break after page/page;

compute after brk;
    line " ";
endcomp;

compute before _page_ / style={just=left protectspecialchars=off};
    line "\b\fs24\sa24&_FSRTITL." ;

    line " ";
    line "&linebot";
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.';
    line 'Note: Percentages are based on the number of subjects indicated in the column header (N).';
    line 'Note: Subjects are counted only once per deviation category but can be counted in more than one deviation category.';
    LINE ' ';
    line 'Appendix 15.3.1.10';
    line "Study ID: ZRHM-REXA-08-US      Program: &TFLprg      Status: &status" &_blankn.*"\-~" "&sysdate" &_blankn.*"\-~" "(Page &i
of &page)";
    endcomp;
run;
%end;
ods rtf close;
ods results on;
ods path sashelp.tmplmst (read);

%mend outrtf;

%outrtf(blankn=36, halfblnk=N);

```

```
ods listing;
```

```
%m_logchk2;
```