

```

/*****
** STUDY ID :                000000106343
**
** PROGRAM NAME :           t_cough.sas
**
** DATE :                   27May2015
**
** PROGRAMMER :             cvn_aramasah
**
** PURPOSE :                QC the table Summary of cough assessments over study- safety
population (t_15_2_6_25)
**
** INPUT DATA :            ADAM.ADSL, ADAM.adqssym
**
** 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_25;
```

```
%let TFLprg=t_cough.sas;
```

```
%let TFL_Part=%scan(&_amp;_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;
```

```
proc sql;
```

```

select count(distinct usubjid) into: pre_ths from adam.adsl(where =(safbfl = "Y" and trt01an = 4));
select count(distinct usubjid) into: pre_mcc from adam.adsl(where =(safbfl = "Y" and trt01an = 5));
select count(distinct usubjid) into: pre_sa from adam.adsl(where =(safbfl = "Y" and trt01an = 3));
select count(distinct usubjid) into: pre_prod from adam.adsl(where =(safbfl = "Y" and trt01an = 96));
select count(distinct usubjid) into: pre_all from adam.adsl(where =(safbfl = "Y" and trt01an in (3 4 5 96)));

```

```
quit;
```

```
proc sql;
```

```

select count(distinct usubjid) into: rand_ths from adam.adsl(where =(safaf1 = "Y" and trt01an = 4));
select count(distinct usubjid) into: rand_mcc from adam.adsl(where =(safaf1 = "Y" and trt01an = 5));
select count(distinct usubjid) into: rand_sa from adam.adsl(where =(safaf1 = "Y" and trt01an = 3));
select count(distinct usubjid) into: rand_all from adam.adsl(where =(safaf1 = "Y" and trt01an in (3 4 5)));

```

```
quit;
```

```
%macro cough(flag=,name=);
```

```
*****,
```

```
* read in data ;
```

```
*****,
```

```
data ex_&name. int_&name. courfeq_&name. sput_&name.;
```

```
    set adam.adqssym;* (where=(anl01fl='Y'));
```

```
    &flag.;
```

```

    if trta='THSm2.2' then trtan=1;

    if trta='mCC' then trtan=2;

    if trta='SA' then trtan=3;

    if trta='Product Test' then trtan=96;

    if missing(trtan) then delete;

    if index(trta,'Expos') then delete;


    if paramn=1 then output ex_&name.;

    if paramn=3 and not missing(aval) then output int_&name.;

    if paramn=4 and not missing(aval) then output courfeq_&name.;

    if paramn=5 and not missing(aval) then output sput_&name.;

run;


data ex01;

    set ex_&name.;

    if missing(avalc) then avalc='Missing';

run;


proc sort data=ex01;

    by subjid descending avalc;

run;


proc freq data=ex01 noprint;

    tables subjid*trtan*avalc / out=ex02(drop=percent);

run;

```

```
proc sort data=ex02;

    by subjid descending avalc;

run;
```

```
data ex03;

    set ex02;

    by subjid descending avalc;

    if first.subjid;

run;
```

```
proc summary data=ex03;

    class avalc trtan;

    var count;

    output out=ex04

    sum(count)=sum;

run;
```

```
data ex04a(where=(avalc ne " and trtan ne .));

    set ex04(keep=avalc trtan sum);

run;
```

```
proc sort data=ex04a;

    by descending avalc;
```

```

run;

proc transpose data=ex04a out=ex_ev02 (drop=_name__label_) prefix=ev;

    by descending avalc;

    id trtan;

    var sum;

run;

data ex_ev03;

    set ex_ev02;

    rown=_n_+1;

    text='Has subject experienced cough in the study period assessed';

    variable=put(avalc,$80.);

run;


proc sort data=ex_ev03;

by rown;

run;

/*Pull out event numbers for bottom half of table*/

data int_ev;

    set int_&name.;

    if missing(aval) then delete;

run;

proc freq data=int_ev;

    table subjid*trtan*avalc*aval / noprint out=int_ev01(drop=percent);

run;

proc sort data=int_ev01;

```

```
        by subjid descending aval;  
run;
```

```
data int_ev01a;  
    set int_ev01;  
    by subjid descending aval;  
    if first.subjid then output;  
run;
```

```
proc summary data=int_ev01a;  
    class aval avalc trtan;  
    var count;  
    output out=int_ev01b  
    sum(count)=sum;  
run;
```

```
data int_ev01c(where=(avalc ne " and aval ne . and trtan ne .));  
    set int_ev01b(keep=aval avalc trtan sum);  
run;
```

```
proc sort data=int_ev01c;  
    by aval avalc;  
run;
```

```
proc transpose data=int_ev01c out=int_ev02 (drop=_name_ _label_) prefix=ev;
```

```
        by aval avalc;

        id trtan;

        var sum;

run;

data int_ev03;

    set int_ev02;

    rown=aval+6;

    text='Intensity of cough';

    variable=put(avalc,$80.);

run;

/*Pull out n numbers for top half*/

proc sort data=ex01 out=ex_sub1;

    by trtan usubjid descending avalc ;

run;

proc sort data=ex_sub1 out=ex_sub2 nodupkey;

    by trtan usubjid descending avalc;

run;

proc sort data=ex_sub2;

    by descending avalc;

run;

proc freq data=ex_sub2 noprint;

    tables trtan*usubjid*avalc / out=ex_sub3(drop=percent rename=(count=count2));
```



```
run;
```

```
proc sort data=ex_sub3;  
    by trtan usubjid descending avalc;
```

```
run;
```

```
proc sort data=ex_sub3 nodupkey out=ex_sub4;  
    by trtan usubjid ;
```

```
run;
```

```
proc freq data=ex_sub4;  
    table trtan*avalc / noprint out=ex_sub01(drop=percent);
```

```
run;
```

```
/* for missing */
```

```
proc sort data=ex_sub01;  
    by trtan;
```

```
run;
```

```
proc transpose data=ex_sub01 out=ex_sub01_trans;  
    by trtan;  
    id avalc;  
    var count;
```

```
run;
```

```

data ex_sub01_trans;

    set ex_sub01_trans;

    _Y_N=sum(y, n/*, Missing*/);

        if trtan=1 and _Y_N ne &pre_ths. then do;

            avalc='Missing';

            count= (&pre_ths. - _Y_N);

        end;

/*
        else count=0;*/

        if trtan=2 and _Y_N ne &pre_mcc. then do;

            avalc='Missing';

            count= (&pre_mcc. - _Y_N);

        end;

        if trtan=3 and _Y_N ne &pre_sa. then do;

            avalc='Missing';

            count= (&pre_sa. - _Y_N);

        end;

        %if &name.=1 %then %do;

            if trtan=96 and _y_n ne &pre_prod. then do;

                avalc='Missing';

                count= (&pre_prod. - _y_n);

            end;

        %end;

    keep trtan count avalc;

run;

```

```

data ex_sub01_;

    set ex_sub01 ex_sub01_trans;

run;


/* missing end */

proc sort data=ex_sub01_;

    by descending avalc;

run;


proc transpose data=ex_sub01_ out=ex_sub02 (drop=_name_ _label_) prefix=sub;

    by descending avalc;

    id trtan;

    var count;

run;

data ex_sub03;

    set ex_sub02;

    if avalc ne "";

    rown=_n_+1;

    text='Has subject experienced cough in the study period assessed';

    variable=put(avalc,$80.);

run;


/* for % other than has subject experienced cough in the study assessed */

```

```

data perc_1_&name.;

set ex_sub03;

if avalc='Y';

place=&name.;

if sub1 ne . then ths_perc_1=sub1;

if sub2 ne . then mcc_perc_1=sub2;

if sub3 ne . then sa_perc_1=sub3;

%if &name.=1 %then %do;

if sub96 ne . then prod_perc_1=sub96;

total_perc=sum(sub1, sub2, sub3, sub96);

drop sub1 sub2 sub3 sub96 avalc;

%end;

%else %do;

total_perc=sum(sub1, sub2, sub3);

drop sub1 sub2 sub3 avalc;

%end;

run;


proc sort data= perc_1_&name.;

by place;

run;


/*Pull out n numbers for bottom third*/

data int_sub;

    set int_&name. (where=(not missing(aval)));

```

```

    by subjid;

    retain subcount maxint;

    maxint=max(maxint,aval);

    if first.subjid then do;

        maxint=aval;

        subcount=subjid;

        end;

    keep subjid aval avalc maxint subcount trtan;

run;

proc sort data=int_sub out=int_sub01;

    by subjid descending aval;

run;

data int_sub02;

    set int_sub01;

    by subjid descending aval;

    if not first.subjid then delete;

run;

proc freq data=int_sub02;

    table trtan*aval*avalc / noprint out=int_sub03;

run;


proc sort data=int_sub03;

    by aval avalc;

run;

```

```
proc transpose data=int_sub03 out=int_sub04 (drop=_name__label_) prefix=sub;
```

```
    by aval avalc;
```

```
    id trtan;
```

```
    var count;
```

```
run;
```

```
data int_sub05;
```

```
    set int_sub04;
```

```
    length variable $80.;
```

```
    rown=aval+6;
```

```
    text='Intensity of cough';
```

```
    variable=put(avalc,$50.);
```

```
run;
```

```
/*FREQUENCY OF COUGH*/
```

```
data courfeq2;
```

```
    set courfeq_&name.;
```

```
    if missing(avalc) then avalc='Missing';
```

```
run;
```

```
proc sort data=courfeq2;
```

```
    by subjid descending aval;
```

```
run;
```

```
proc freq data=courfeq2 noprint;
```

```
    tables subjid*trtan*aval*avalc / out=courfeq3(drop=percent);
```

```
run;
```

```
proc sort data=courfeq3;  
    by subjid descending aval;  
run;
```

```
data courfeq4;  
    set courfeq3;  
    by subjid descending aval;  
    if first.subjid;  
run;
```

```
proc summary data=courfeq4;  
    class aval avalc trtan;  
    var count;  
    output out=courfeq5  
    sum(count)=sum;  
run;
```

```
data courfeq6(where=(avalc ne " and trtan ne . and aval ne .));  
    set courfeq5(keep=avalc trtan sum aval);  
run;
```

```
proc sort data=courfeq6;  
    by aval avalc;
```

```
run;
```

```
proc transpose data=courfeq6 out=courfeq7 (drop=_name__label_) prefix=ev;
```

```
    by aval avalc;
```

```
    id trtan;
```

```
    var sum;
```

```
run;
```

```
data courfeq8;/* THIS IS NUMBER OF EVENTS FOR SOUGH FREQUENCY */
```

```
    set courfeq7;
```

```
    rown=aval+13;
```

```
    text='Frequency of cough';
```

```
    variable=strip(put(avalc,$80.));
```

```
run;
```

```
data courfeqsub;
```

```
    set courfeq_&name.(where=(not missing(aval)));
```

```
    by subjid;
```

```
    retain subcount maxint;
```

```
    maxint=max(maxint,aval);
```

```
    if first.subjid then do;
```

```
        maxint=aval;
```

```
        subcount=subjid;
```

```
    end;
```

```
    keep subjid aval avalc maxint subcount trtan;
```



```

run;

proc sort data=courfeqsub out=courfeqsub1;

    by subjid descending aval;

run;

data courfeqsub2;

    set courfeqsub1;

    by subjid descending aval;

    if not first.subjid then delete;

run;

proc freq data=courfeqsub2;

    table trtan*aval*avalc / noprint out=courfeqsub3;

run;

proc sort data=courfeqsub3;

    by aval avalc;

run;

proc transpose data=courfeqsub3 out=courfeqsub4 (drop=_name_ _label_) prefix=sub;

    by aval avalc;

    id trtan;

    var count;

run;

data courfeqsub5; /*THIS IS NUMBER OF SUBJECTS FOR COUGH FREQUENCY*/

    set courfeqsub4;

    length variable $80.;

```

```
        rown=aval+13;

        text='Frequency of cough';

        variable=strip(put(avalc,$50.));

run;


/*SPUTUM*/

data sput2;

        set sput_&name.;

        if missing(avalc) then avalc='Missing';

run;


proc sort data=sput2;

        by subjid descending aval;

run;


proc freq data=sput2 noprint;

        tables subjid*trtan*aval*avalc / out=sput3(drop=percent);

run;


proc sort data=sput3;

        by subjid descending aval;

run;


data sput4;

        set sput3;
```

```
    by subjid descending aval;  
    if first.subjid;  
run;
```

```
proc summary data=sput4;  
    class aval avalc trtan;  
    var count;  
    output out=sput5  
    sum(count)=sum;  
run;
```

```
data sput6(where=(avalc ne " " and trtan ne . and aval ne .));  
    set sput5(keep=avalc aval trtan sum);  
run;
```

```
proc sort data=sput6;  
    by aval avalc;  
run;
```

```
proc transpose data=sput6 out=sput7 (drop=_name__label_) prefix=ev;  
    by aval avalc;  
    id trtan;  
    var sum;  
run;
```

```
data sput8;/* THIS IS NUMBER OF EVENTS FOR SPUTUM */
```

```
    set sput7;
```

```
    rown=aval+21;
```

```
    text='Amount of sputum produced';
```

```
    variable=put(avalc,$80.);
```

```
run;
```

```
data sputsub;
```

```
    set sput_&name. (where=(not missing(aval)));
```

```
    by subjid;
```

```
    retain subcount maxint;
```

```
    maxint=max(maxint,aval);
```

```
    if first.subjid then do;
```

```
        maxint=aval;
```

```
        subcount=subjid;
```

```
    end;
```

```
    keep subjid aval avalc maxint subcount trtan;
```

```
run;
```

```
proc sort data=sputsub out=sputsub1;
```

```
    by subjid descending aval;
```

```
run;
```

```
data sputsub2;
```

```
    set sputsub1;
```

```
    by subjid descending aval;
```

```
    if not first.subjid then delete;
```

```

run;

proc freq data=sputsub2;

    table trtan*aval*avalc / noprint out=sputsub3;

run;


proc sort data=sputsub3;

    by aval avalc;

run;

proc transpose data=sputsub3 out=sputsub4 (drop=_name__label_) prefix=sub;

    by aval avalc;

    id trtan;

    var count;

run;

data sputsub5;/*THIS IS NUMBER OF SUBJECTS FOR SPUTUM*/

    set sputsub4;

    length variable $80.;

    rown=aval+21;

    text='Amount of sputum produced';

    variable=put(avalc,$50.);

run;


data rows;

    length variable $80.;

sect=1;

```

```

    rown=1; variable='Has subject experienced cough in the study period assessed?';

    output;

    rown=2; variable='Yes';

    output;

    rown=3; variable='No';

    output;

sect=2;

    rown=6;      variable='Intensity of cough';

    output;

    rown=7;      variable='Very mild';

    output;

    rown=8; variable='Mild';

    output;

    rown=9; variable='Moderate';

    output;

    rown=10; variable='Severe';

    output;

    rown=11; variable='Very severe';

    output;

/*    rown=12; variable='Missing';*/

/*    output;*/

sect=3;

    rown=13; variable='Frequency of cough';

    output;

    rown=14; variable='Rarely';

```

```
output;

rown=15; variable='Sometimes';

output;

rown=16; variable='Fairly often';

output;

rown=17; variable='Often';

output;

rown=18; variable='Almost always';

output;

/*   rown=19; variable='Missing';*/

/*   output;*/

sect=4;

rown=20; variable='Amount of sputum produced';

output;

rown=21; variable='No sputum';

output;

rown=22; variable='A moderate amount of sputum';

output;

rown=23; variable='A large amount of sputum';

output;

rown=24; variable='A very large amount of sputum';

output;

/*   rown=25; variable='Missing';*/

/*   output;*/

run;
```

```
/* n of the 2,3,4 sections of the table */
```

```
proc freq data=int_sub02;  
    table trtan / noprint out=int_sub03_n;  
run;
```

```
proc freq data=courfeqsub2;  
    table trtan / noprint out=courfeqsub3_n;  
run;
```

```
proc freq data=sputsub2;  
    table trtan / noprint out=sputsub3_n;  
run;
```

```
proc transpose data=int_sub03_n out=int_sub03_n_1;  
id trtan;  
var count;  
run;
```

```
data int_sub03_n_1;  
length item $50.;  
set int_sub03_n_1;
```



```
item='n';
```

```
run;
```

```
data int_sub03_n_1;
```

```
length item $50.;
```

```
set int_sub03_n_1;
```

```
item='n';
```

```
drop _name__label_;
```

```
rown=6.5;
```

```
%if &name.=2 %then %do; all=sum(_1, _2, _3); %end;
```

```
%if &name.=1 %then %do; all=sum(_1, _2, _3, _96); %end;
```

```
run;
```

```
proc transpose data=courfeqsub3_n out=courfeqsub3_n_1;
```

```
id trtan;
```

```
var count;
```

```
run;
```

```
data courfeqsub3_n_1;
```

```
length item $50.;
```

```
set courfeqsub3_n_1;
```

```
item='n';
```

```
drop _name__label_;
```

```
rown=13.5;
```

```
%if &name.=2 %then %do; all=sum(_1, _2, _3); %end;
```

```
%if &name.=1 %then %do; all=sum(_1, _2, _3, _96); %end;
```

```
run;
```

```
proc transpose data=sputsub3_n out=sputsub3_n_1;
```

```
id trtan;
```

```
var count;
```

```
run;
```

```
data sputsub3_n_1;
```

```
length item $50.;
```

```
set sputsub3_n_1;
```

```
item='n';
```

```
drop _name_ _label_;
```

```
rown=20.5;
```

```
%if &name.=2 %then %do; all=sum(_1, _2, _3); %end;
```

```
%if &name.=1 %then %do; all=sum(_1, _2, _3, _96); %end;
```

```
run;
```

```
/* missing */
```

```
/* section 1 */
```

```
proc sort data=int_sub03;
```

```
by trtan;
```

```
run;
```

```
proc transpose data=int_sub03 out=int_sub03_trans;
```

```

        by trtan;

        id avalc;

        var count;

run;

data int_sub03_trans;

length item $50.;

        set int_sub03_trans;

        _Y_N=sum(VERY_MILD, MILD,MODERATE, /*VERY_SEVERE,*/ SEVERE/*, Missing*/);

                item='n';

        keep item trtan _Y_N ;

run;

proc transpose data=int_sub03_trans out=int_sub03_trans_suffix=_;

by item;

id trtan;

var _Y_N;

run;

proc sort data=int_sub03_n_1;

by item;

proc sort data=int_sub03_trans_;

by item;

run;

```

```

data int_sub03_n_1_;

merge int_sub03_n_1 int_sub03_trans_ (drop=_name_);

by item;

if _1 ne _1_ then do; item='Missing'; _1=_1 - _1_; end;

if _2 ne _2_ then do; item='Missing'; _2=_2 - _2_; end;

if _3 ne _3_ then do; item='Missing'; _3=_3 - _3_; end;

%if &name.=1 %then %do;

if _96 ne _96_ then do; item='Missing'; _96=_96 - _96_; end;

drop _1_ _2_ _3_ _96_;

%end;

drop _1_ _2_ _3_;

run;

/* section 2 */

proc sort data=courfeqsub3;

    by trtan;

run;

proc transpose data=courfeqsub3 out=courfeqsub3_trans;

    by trtan;

    id avalc;

    var count;

run;

data courfeqsub3_trans;

```

```
length item $50.;
```

```
set courfeqsub3_trans;
```

```
_Y_N=sum(RARELY, SOMETIMES, FAIRLY_OFTEN, OFTEN/*, ALMOST_ALWAYS, Missing*/);
```

```
item='n';
```

```
keep item trtan _Y_N ;
```

```
run;
```

```
proc transpose data=courfeqsub3_trans out=courfeqsub3_trans_ suffix=_;
```

```
by item;
```

```
id trtan;
```

```
var _Y_N;
```

```
run;
```

```
proc sort data=courfeqsub3_n_1;
```

```
by item;
```

```
proc sort data=courfeqsub3_trans_;
```

```
by item;
```

```
run;
```

```
data courfeqsub3_n_1_;
```

```
merge courfeqsub3_n_1 courfeqsub3_trans_ (drop=_name_);
```

```
by item;
```

```
if _1 ne _1_ then do; item='Missing'; _1=_1 - _1_; end;
```

```
if _2 ne _2_ then do; item='Missing'; _2=_2 - _2_; end;
```

```
if _3 ne _3_ then do; item='Missing'; _3=_3 - _3_; end;

drop _1_ _2_ _3_;

%if &name.=1 %then %do;

if _96 ne _96_ then do; item='Missing'; _96=_96 - _96_; end;

drop _1_ _2_ _3_ _96_;

%end;

run;

/* section 3 */
```

```
proc sort data=sputsub3;

    by trtan;

run;
```

```
proc transpose data=sputsub3 out=sputsub3_trans;

    by trtan;

    id avalc;

    var count;

run;
```

```
data sputsub3_trans;

length item $50.;

    set sputsub3_trans;

    _Y_N=sum(NO_SPUTUM, A_MODERATE_AMOUNT_OF_SPUTUM,
A_LARGE_AMOUNT_OF_SPUTUM/*, A_VERY_LARGE_AMOUNT_OF_SPUTUM, Missing*/);

    item='n';

    keep item trtan _Y_N;
```

```
run;
```

```
proc transpose data=sputsub3_trans out=sputsub3_trans_ suffix=_;
```

```
by item;
```

```
id trtan;
```

```
var _Y_N;
```

```
run;
```

```
proc sort data=sputsub3_n_1;
```

```
by item;
```

```
proc sort data=sputsub3_trans_;
```

```
by item;
```

```
run;
```

```
data sputsub3_n_1_;
```

```
merge sputsub3_n_1 sputsub3_trans_ (drop=_name_);
```

```
by item;
```

```
if _1 ne _1_ then do; item='Missing'; _1=_1 - _1_; end;
```

```
if _2 ne _2_ then do; item='Missing'; _2=_2 - _2_; end;
```

```
if _3 ne _3_ then do; item='Missing'; _3=_3 - _3_; end;
```

```
drop _1_ _2_ _3_;
```

```
%if &name.=1 %then %do;
```

```
if _96 ne _96_ then do; item='Missing'; _96=_96 - _96_; end;
```

```
drop _1_ _2_ _3_ _96_;
```

```
%end;
```

```
run;
```

```
/* all together */
```

```
data all_missing;
```

```
set int_sub03_n_1_ (in=sec_2) courfeqsub3_n_1_ (in=sec_3) sputsub3_n_1_ (in=sec_4);
```

```
if sec_2 and item='Missing' then rown=12; else delete;
```

```
if sec_3 and item='Missing' then rown=19; else delete;
```

```
if sec_4 and item='Missing' then rown=25; else delete;
```

```
run;
```

```
/* end */
```

```
proc sort data=ex_sub03;
```

```
by text rown variable avalc;
```

```
proc sort data=ex_ev03;
```

```
by text rown variable avalc;
```

```
proc sort data=int_sub05;
```

```
by text rown variable avalc aval;
```

```
proc sort data=int_ev03;
```

```
by text rown variable avalc aval;
```

```
proc sort data=sputsub5;
```

```
by text rown variable avalc aval;
```

```
proc sort data=sput8;
```

```
by text rown variable avalc aval;
```



```
proc sort data=courfeqsub5;  
    by text rown variable avalc aval;  
proc sort data=courfeq8;  
    by text rown variable avalc aval;  
run;
```

```
data part_1;  
    merge ex_sub03 ex_ev03;  
    by text rown variable avalc;  
    paramn=1;  
run;
```

```
data part_2;  
    merge int_sub05 int_ev03;  
    by text rown variable avalc aval;  
run;
```

```
data part_3;  
    merge courfeqsub5 courfeq8;  
    by text rown variable avalc aval;  
run;
```

```
data part_4;  
    merge sputsub5 sput8;  
    by text rown variable avalc aval;
```

```
run;
```

```
data part_2_;
```

```
set part_2 int_sub03_n_1;
```

```
paramn=2;
```

```
run;
```

```
data part_3_;
```

```
set part_3 courfeqsub3_n_1;
```

```
paramn=3;
```

```
run;
```

```
data part_4_;
```

```
set part_4 sputsub3_n_1;
```

```
paramn=4;
```

```
run;
```

```
/* setting all sections of the table and missing */
```

```
data together;
```

```
    set part_1(in=a) part_2_ (in=b) part_3_ part_4_ all_missing;
```

```
    drop text;
```

```
    if compress(variable)='Y' then variable='Yes';
```

```
    if compress(variable)='N' then variable='No';
```

```
    if _1 ne . then sub1=_1;
```

```
    if _2 ne . then sub2=_2;
```

```

    if _3 ne . then sub3=_3;

    if item ne " then variable=item;

    %if &name.=1 %then %do; if _96 ne . then sub96=_96; %end;

run;


proc sort data=together;

    by rown variable;

run;


data table;

    merge together rows;

    by rown variable;

    length variable1 variable2 $200.;

    if rown=4 and variable='Missing' then go to zero;

    else if rown=4 and variable ne 'Missing' then delete;

    else if rown in(2,3,7,8,9,10,11, 14, 15, 16, 17, 18, 21, 22, 23, 24) then do;

        zero: if missing(ev1) then ev1=0;

        if missing(ev2) then ev2=0;

        if missing(ev3) then ev3=0;

        if missing(ev96) then ev96=0;

        if missing(sub1) then sub1=0;

        if missing(sub2) then sub2=0;

        if missing(sub3) then sub3=0;

        if missing(sub96) then sub96=0;

    end;

```

```

        if rown not in (1, 6,13, 20) then do;

        ov_sub=sum(sub1, sub2, sub3, sub96);

        if rown not in (6.5, 13.5 20.5) then do;

        ov_ev=sum(ev1, ev2, ev3, ev96);

        end;

        end;

/*      end;*/

attrib wrap length = $200;

wrap =variable;


        if rown in(2,3,4,7,8,9,10,11, 14, 15, 16, 17, 18, 21, 22, 23, 24) then
variable1="$S={foreground=white} . $S={}" || wrap ;

        if rown in (22, 23, 24) then variable1=tranwrd(variable1,'of sputum',"$n $S={foreground=white} .
$S={}" || 'of sputum');

        variable2=variable1;

        if variable2="" and variable ne "" then variable2=variable;

        bynum=1;

        drop aval variable1 variable2 wrap;

/*      if avalc ne "" then variable=' ' || strip(variable);*/

        if rown not in (1 6 13 20) then variable=' ' || strip(variable);

        else variable=strip(variable);

place=&name.;

run;

```

```
proc sort data=table;
```

```
    by place;
```

```
run;
```

```
data table01_;
```

```
    merge table perc_1_&name. (drop=row variable);
```

```
    by place;
```

```
run;
```

```
data table01_&name.;
```

```
    set table01_;
```

```
%if &name.=1 %then %do;
```

```
    attrib  sub1 sub2 sub3 sub96 ov_sub label="n"
```

```
           p1 p2 p3 p96 ov_p label='(%)' length=$8.
```

```
           ev1 ev2 ev3 ev96 ov_ev label="Events";
```

```
%end;
```

```
%if &name.=2 %then %do;
```

```
    attrib  sub1 sub2 sub3 ov_sub label="n"
```

```
           p1 p2 p3 ov_p label='(%)' length=$8.
```

```
           ev1 ev2 ev3 ov_ev label="Events";
```

```
%end;
```

```
%if &name.=1 %then %do;
```

```
if rown not in (6.5 13.5 20.5) then do;
```

```
    if rown in (2 3 4) then do;
```

```
        if sub1>0 then p1a=(sub1/&pre_ths.)*100;
```

```

    if sub2>0 then p2a=(sub2/&pre_mcc.)*100;

    if sub3>0 then p3a=(sub3/&pre_sa.)*100;

    if sub96>0 then p96a=(sub96/&pre_prod.)*100;

    if ov_sub>0 then ov_pa=(ov_sub/&pre_all.)*100;

    end;

    else do;

    if sub1>0 then p1a=(sub1/THS_PERC_1)*100;

    if sub2>0 then p2a=(sub2/MCC_PERC_1)*100;

    if sub3>0 then p3a=(sub3/SA_PERC_1)*100;

    if sub96>0 then p96a=(sub96/prod_perc_1)*100;

    if ov_sub>0 then ov_pa=(ov_sub/total_perc)*100;

    end;

end;

%end;

%if &name.=2 %then %do;

if rown not in (6.5 13.5 20.5) then do;

    if rown in (2 3 4) then do;

    if sub1>0 then p1a=(sub1/&rand_ths.)*100;

    if sub2>0 then p2a=(sub2/&rand_mcc.)*100;

    if sub3>0 then p3a=(sub3/&rand_sa.)*100;

    if ov_sub>0 then ov_pa=(ov_sub/&rand_all.)*100;

    end;

    else do;

    if sub1>0 then p1a=(sub1/THS_PERC_1)*100;

    if sub2>0 then p2a=(sub2/MCC_PERC_1)*100;

```

```

if sub3>0 then p3a=(sub3/SA_PERC_1)*100;

if ov_sub>0 then ov_pa=(ov_sub/total_perc)*100;

end;

end;

%end;


if p1a=100 then p1=trim(''|compress(put(p1a,8.))||'%');

else if not missing(p1a) and p1a>=10 and p1a<100 then
p1=trim(''|compress(put(p1a,8.1))||'%'); /* 1) JH 22OCT2014 */

else if not missing(p1a) and p1a<10 then p1=trim(''|compress(put(p1a,8.1))||'%');


if p2a=100 then p2=trim(''|compress(put(p2a,8.))||'%');

else if not missing(p2a) and p2a>=10 and p2a<100 then
p2=trim(''|compress(put(p2a,8.1))||'%'); /* 1) JH 22OCT2014 */

else if not missing(p2a) and p2a<10 then p2=trim(''|compress(put(p2a,8.1))||'%');


if p3a=100 then p3=trim(''|compress(put(p3a,8.))||'%');

else if not missing(p3a) and p3a>=10 and p3a<100 then
p3=trim(''|compress(put(p3a,8.1))||'%'); /* 1) JH 22OCT2014 */

else if not missing(p3a) and p3a<10 then p3=trim(''|compress(put(p3a,8.1))||'%');

%if &name.=1 %then %do;

if p96a=100 then p96=trim(''|compress(put(p96a,8.))||'%');

else if not missing(p96a) and p96a>=10 and p96a<100 then
p96=trim(''|compress(put(p96a,8.1))||'%'); /* 1) JH 22OCT2014 */

else if not missing(p96a) and p96a<10 then p96=trim(''|compress(put(p96a,8.1))||'%');

%end;

if ov_pa=100 then ov_p=trim(''|compress(put(ov_pa,8.))||'%');

```

```

        else if not missing(ov_pa) and ov_pa>=10 and ov_pa<100 then
ov_p=trim(''|compress(put(ov_pa,8.1))||'%'); /* 1) JH 22OCT2014 */

        else if not missing(ov_pa) and ov_pa<10 then ov_p=trim(''|compress(put(ov_pa,8.1))||'%');

run;

%mend cough;

%cough(flag=where safbfl='Y' and asper=1, name=1);

%cough(flag=where safaf1='Y' and asper gt 1, name=2);

data table;

length category $50.;

set table01_1 table01_2;

    if place=1 then category="Pre-Randomization Period";

    else if place=2 then category="Randomized Period";

    drop avalc;

    if sub1 not in (0 .) then ths_num=strip(put(sub1,best.))||' '|strip(p1); else
ths_num=strip(put(sub1,best.));

    if sub2 not in (0 .) then mcc_num=strip(put(sub2,best.))||' '|strip(p2); else
mcc_num=strip(put(sub2,best.));

    if sub3 not in (0 .) then sa_num=strip(put(sub3,best.))||' '|strip(p3); else
sa_num=strip(put(sub3,best.));

```



```
if sub96 not in (0 .) then PRODUCT_TEST_num=strip(put(sub96,best.))||' '||strip(p96); else  
PRODUCT_TEST_num=strip(put(sub96,best.));
```

```
if ov_sub not in (0 .) then overall_num=strip(put(ov_sub,best.))||' '||strip(ov_p); else  
overall_num=strip(put(ov_sub,best.));
```

```
if compress(variable)='Y' then variable='Has subject experienced cough in the study period assessed?';
```

```
/* drop sub1 sub2 sub3 sub96 p1 p2 p3 p96 ov_p; */
```

```
if place=1 and variable='Has subject experienced cough in the study period assessed?' then do;  
paramn=1;rown=1;page=1;retain page paramn;end;
```

```
if place=1 and variable='Intensity of cough' then paramn=2;retain paramn;
```

```
if place=1 and variable='Frequency of cough' then do; paramn=3; page=2;retain page  
paramn;end;
```

```
if place=1 and variable='Amount of sputum produced' then do;paramn=4;page=3;retain page  
paramn;end;
```

```
if place=2 and variable='Has subject experienced cough in the study period assessed?' then  
do;paramn=1; rown=1;page=4;retain page paramn;end;
```

```
if place=2 and variable='Intensity of cough' then paramn=2;retain paramn;
```

```
if place=2 and variable='Frequency of cough' then do; paramn=3; page=5;retain page  
paramn;end;
```

```
if place=2 and variable='Amount of sputum produced' then do; paramn=4; page=6;retain page  
paramn;end;
```

```
maxpage=6;
```

```
if EV1 ne . then ths_eve=strip(put(EV1,best.));
```

```
if EV2 ne . then mcc_eve=strip(put(EV2,best.));
```

```
if EV3 ne . then sa_eve=strip(put(EV3,best.));
```

```
if EV96 ne . then prod_eve=strip(put(EV96,best.));
```

```
if ov_ev ne . then all_eve=strip(put(ov_ev,best.));
```

```
if place=2 then do;prod_eve="";PRODUCT_TEST_num="";end;
```

```
if compress(variable)='Verysevere' then paramn=2;
```

```

        if compress(variable)='Almostalways' then paramn=3;

        if compress(variable)='Averylargeamountofsputum' then paramn=4;

run;

proc sql noprint;

    create table tfls.&tflno as

        select category, variable, ths_num as ths_count, ths_eve as ths_events, mcc_num as
mcc_count, mcc_eve as mcc_events,

            sa_num as sa_count, sa_eve as sa_events, PRODUCT_TEST_num as product_test_count,
prod_eve as product_test_events,

            overall_num as overall_count, all_eve as overall_events, page, rown, paramn

        from table

        order by place, rown, sect;

quit;

data paging;

    set table;

    by place rown sect;

/*  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.)));

    call symput("maxpage",compress(put(maxpage,best.)));

```

```

        flag=1;

run;

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;


%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 footery=1440 ;

ods noproctitle;


%do i=1 %to &page;

title ;

footnote;

```

```

%let wd=0;

%let supfl=0;


%let npage=%eval(&i);


data comp;

    set paging end=eof;

        where page=&i;

            call symput("place",compress(put(place,best.)));


/* Amend title as needed */
        _firtitl="Table 15.2.6.25 Summary of Cough Assessments Over Study - 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('perid', strip(category));

                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 place page rown /*with paramn PARAMC*/variable /*avisitn ord visit statord stat*/

%if &place.=2 %then %do;

("THSm2.2\$(N=%cmpres(&rand_ths.)) &linebot" THS_NUM ths_eve)

("mCC\$(N=%cmpres(&rand_mcc.)) &linebot" MCC_NUM mcc_eve)

("SA\$(N=%cmpres(&rand_sa.)) &linebot" SA_NUM sa_eve)

("Overall Safety\$(N=%cmpres(&rand_all.)) &linebot" OVERALL_NUM
all_eve)

%end;

%else %if &place.=1 %then %do;

("THSm2.2\$(N=%cmpres(&pre_ths.)) &linebot" THS_NUM ths_eve)

("mCC\$(N=%cmpres(&pre_mcc.)) &linebot" MCC_NUM mcc_eve)

("SA\$(N=%cmpres(&pre_sa.)) &linebot" SA_NUM sa_eve)

("Product Test\$Only\$(N=%cmpres(&pre_prod.)) &linebot"
PRODUCT_TEST_NUM prod_eve)

("Overall\$Safety\$(N=%cmpres(&pre_all.)) &linebot" OVERALL_NUM
all_eve)

%end;;

;

```
define flag      / order order=internal noprint;

define place      / order order=internal noprint;

define page      / order order = internal noprint;

define rown      / order order = internal noprint;

define variable   / display style={just=l cellwidth=0.9 cm}
style(header)={just=left}'Variable'; /* 11) JMH 16Sep2014 */

%if &place.=2 %then %do;

        define THS_NUM      / display style={just=center cellwidth=0.3 cm}
style(header)={just=center} 'n(%)';

        define ths_eve      / display style={just=center cellwidth=0.2 cm}
style(header)={just=center} 'Events';

        define MCC_NUM      / display style={just=center cellwidth=0.3 cm}
style(header)={just=center} 'n(%)';

        define mcc_eve      / display style={just=center cellwidth=0.2 cm}
style(header)={just=center} 'Events';

        define SA_NUM      / display style={just=center cellwidth=0.3 cm}
style(header)={just=center} 'n(%)';

        define sa_eve      / display style={just=center cellwidth=0.2 cm}
style(header)={just=center} 'Events';

        define OVERALL_NUM      / display style={just=center cellwidth=0.3 cm}
style(header)={just=center} 'n(%)';

        define all_eve      / display style={just=center cellwidth=0.2 cm}
style(header)={just=center} 'Events';

%end;

%else %if &place.=1 %then %do;

        define THS_NUM      / display style={just=center cellwidth=0.25 cm}
style(header)={just=center} 'n(%)';

        define ths_eve      / display style={just=center cellwidth=0.17 cm}
style(header)={just=center} 'Events';
```

```

define MCC_NUM      / display style={just=center cellwidth=0.25 cm}
style(header)={just=center} 'n(%)';

define mcc_eve      / display style={just=center cellwidth=0.17 cm}
style(header)={just=center} 'Events';

define SA_NUM      / display style={just=center cellwidth=0.25 cm}
style(header)={just=center} 'n(%)';

define sa_eve      / display style={just=center cellwidth=0.17 cm}
style(header)={just=center} 'Events';

define PRODUCT_TEST_NUM      / display style={just=center cellwidth=0.25
cm} style(header)={just=center} 'n(%)';

define prod_eve      / display style={just=center cellwidth=0.17 cm}
style(header)={just=center} 'Events';

define OVERALL_NUM      / display style={just=center cellwidth=0.3 cm}
style(header)={just=center} 'n(%)';

define all_eve      / display style={just=center cellwidth=0.17 cm}
style(header)={just=center} 'Events';

%end;

```

```

compute variable;

```

```

if substr(variable,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 ord;*/
```

```
/*          line " ";*/
```

```
/*      endcomp;*/
```

```
compute before page / style={protectspecialchars=off};
```

```
    line "&linetop";
```

```
endcomp;
```

```
compute before _page_ / style={just=left protectspecialchars=off};
```

```
%if &place.=1 %then %do;
```

```
    line "\b\fs24\sa24&_FSRTITL." ;
```

```
        line "&linebot";
```

```
        line "Safety Time Period: Pre-Randomization Period";
```

```
%end;
```

```
%else %do;
```

```
    line "\b\fs24\sa24&_FSRTITL." ;
```

```
        line "&linebot";
```

```
        line "Safety Time Period: Randomization Period";
```

```
%end;
```

```
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. The Overall Safety refers to all subjects exposed to THSm2.2.";
```

```
    line "Note: mCC = Menthol conventional cigarettes; SA = Smoking abstinence; THSm2.2  
= Tobacco Heating System 2.2 Menthol.";
```


line "Note: Percentages for 'Has the subject experienced a cough' are based on the number of subjects indicated in the column header (N). Percentages for intensity of cough are based on the number of subjects who have experienced a cough.";

line "Note: Cough experienced in the previous 24h by the subjects is assessed in the morning of Day 0 to Day 6. If subject has answered question more than once then the most severe intensity is presented.";

line "";

line "Appendix 15.3.6.23";

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(blankn=40, halfblk=N);

ods listing;

```
ods listing close;
```

```
proc datasets lib=work nolist memtype=data kill; quit;
```

```
%m_logchk;
```

```
*=====;
```

```
* END OF PROGRAM CODE                ;
```

```
*=====;
```