

```

%_mprintto;
options notes nosource;
proc datasets lib=work nolist memtype=data kill; quit;
%put NOTE:
=====;
%put NOTE: Covance Study Number : 000000106326;
%put NOTE: Client Protocol ID   : ZRHM-PK-05-JP;
%put NOTE: Program Name        : t_cough.sas;
%put NOTE: Purpose              : table of cough assessments;
%put NOTE: ;
%put NOTE: Input Data           : ADAM.ADQSSYM ADAM.ADSL;
%put NOTE: Output               : t_15_2_6_18(cough);
%put NOTE: Macros Called        : _MPRINTTO;
%put NOTE: ;
%put NOTE: Programmed by        : cvn_aobyrne;
%put NOTE: Creation Date        : 2014-08-06;
%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: ;
%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_18(cough);

/* 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 ;
*****;
data ex int courfeq sput;
    set adam.adqssym(where=(saffl='Y'));
    if missing(trtseqan) then delete;
    if index(trtseqa,'Expos') then delete;
    if paramn=1 then output ex;
    if paramn=3 and not missing(aval) then output int;
    if paramn=4 and not missing(aval) then output courfeq;
    if paramn=5 and not missing(aval) then output sput;
run;
/*Pull out event numbers for top half of table*/
data ex01;
    set ex;
    if missing(avalc) then avalc='Missing';
run;

proc sort data=ex01;
    by subjid descending avalc;
run;

proc freq data=ex01 noprint;
    tables subjid*trtseqan*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 trtseqan;
    var count;
    output out=ex04
    sum(count)=sum;
run;

data ex04a(where=(avalc ne '' and trtseqan ne .));
    set ex04(keep=avalc trtseqan 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 trtseqan;
    var sum;
run;

data ex_ev03;
    set ex_ev02;
    rown=_n_+1;
    variable=put(avalc,$80.);
run;

/*Pull out event numbers for bottom half of table*/
data int_ev;
    set int;
    if missing(aval) then delete;
run;

proc freq data=int_ev;
    table subjid*trtseqan*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 trtseqan;
    var count;
    output out=int_ev01b
    sum(count)=sum;
run;

data int_ev01c(where=(avalc ne '' and aval ne . and trtseqan ne .));
    set int_ev01b(keep=aval avalc trtseqan 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 trtseqan;
    var sum;

```

```

run;

data int_ev03;
    set int_ev02;
    rown=aval+6;
    variable=put(avalc,$80.);
run;

/*Pull out n numbers for top half*/
proc sort data=ex01 out=ex_sub1;
    by trtseqan usubjid descending avalc ;
run;

proc sort data=ex_sub1 out=ex_sub2 nodupkey;
    by trtseqan usubjid descending avalc;
run;

proc sort data=ex_sub2;
    by descending avalc;
run;

proc freq data=ex_sub2 noprint;
    tables trtseqan*usubjid*avalc / out=ex_sub3(drop=percent
rename=(count=count2));
run;

proc sort data=ex_sub3;
    by trtseqan usubjid descending avalc;
run;

proc sort data=ex_sub3 nodupkey out=ex_sub4;
    by trtseqan usubjid ;
run;

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

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 trtseqan;
    var count;
run;

data ex_sub03;
    set ex_sub02;
    rown=_n_+1;
    variable=put(avalc,$80.);
run;

```

```

/*Pull out n numbers for bottom third*/
data int_sub;
    set int(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 trtseqan;
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 trtseqan*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 trtseqan;
    var count;
run;

data int_sub05;
    set int_sub04;
    length variable $80.;
    rown=aval+6;
    variable=put(avalc,$50.);
run;

/*FREQUENCY OF COUGH*/
data courfeq2;
    set courfeq;
    if missing(avalc) then avalc='Missing';
run;

proc sort data=courfeq2;
    by subjid descending aval;

```

```

run;

proc freq data=courfeq2 noprint;
    tables subjid*trtseqan*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 trtseqan;
    var count;
    output out=courfeq5
        sum(count)=sum;
run;

data courfeq6(where=(avalc ne '' and trtseqan ne . and aval ne .));
    set courfeq5(keep=avalc trtseqan 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 trtseqan;
    var sum;
run;

data courfeq8; /* THIS IS NUMBER OF EVENTS FOR SOUGH FREQUENCY */
    set courfeq7;
    rown=aval+13;
    variable=put(avalc,$80.);
run;

data courfeqsub;
    set courfeq(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 trtseqan;

```

```

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 trtseqan*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 trtseqan;
    var count;
run;

data courfeqsub5; /*THIS IS NUMBER OF SUBJECTS FOR COUGH FREQUENCY*/
    set courfeqsub4;
    length variable $80.;
    rown=aval+13;
    variable=put(avalc,$50.);
run;

/*SPUTUM*/
data sput2;
    set sput;
    if missing(avalc) then avalc='Missing';
run;

proc sort data=sput2;
    by subjid descending aval;
run;

proc freq data=sput2 noprint;
    tables subjid*trtseqan*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 trtseqan;
    var count;
    output out=sput5
    sum(count)=sum;
run;

data sput6(where=(avalc ne '' and trtseqan ne . and aval ne .));
    set sput5(keep=avalc aval trtseqan 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 trtseqan;
    var sum;
run;

data sput8; /* THIS IS NUMBER OF EVENTS FOR SPUTUM */
    set sput7;
    rown=aval+21;
    variable=put(avalc,$80.);
run;

data sputsub;
    set sput(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 trtseqan;
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 trtsegan*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 trtsegan;
    var count;
run;

data sputsub5; /*THIS IS NUMBER OF SUBJECTS FOR SPUTUM*/
    set sputsub4;
    length variable $80.;
    rown=aval+21;
    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;
    rown=4;    variable='';
    output;
    rown=5;    variable='';
    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='';
    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='';
output;
sect=4;
rown=20; variable='Amount of sputum';
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;
run;

data table;
  merge ex_ev03 ex_sub03 int_ev03 int_sub05 sputsub5 sput8
courfeqsub5 courfeq8 rows;
  by rown;
  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(ev4) then ev4=0;
    if missing(ev5) then ev5=0;
    if missing(sub1) then sub1=0;
    if missing(sub2) then sub2=0;
    if missing(sub3) then sub3=0;
    if missing(sub4) then sub4=0;
    if missing(sub5) then sub5=0;
  end;
  ov_sub=sub1+sub2+sub3+sub4+sub5;
  ov_ev=ev1+ev2+ev3+ev4+ev5;attrib wrap length = $200;
wrap =variable;
if rown not in(1 5 6 12 13 19 20) then do;
  i=30;
  if length(wrap)>i then
    do;
      fin=0;
      do while(fin=0);
        if substr(wrap,i,1)=' ' then
          do;

```

```

wrap=substr(wrap,1,i-1) || "$n
$S={foreground=white} . $S={}" || substr(wrap,i+1);
fin=1;
end;
else i=i-1;
end;
end;
variable="$S={foreground=white} . $S={}"||wrap;
end;
bynum=1;
run;

data adsl;
set adam.adsl(where=(trtseql ne 6 and not missing(trtseql)));
output;
trtseql=99;
trtseql='Overall Randomized';
output;
run;

proc freq data=adsl noprint;
table trtseql*trtseql/ out =tot(drop=percent);
run;

data dumtrts;
attrib trtseql length=$200. trtseql length=8.;
dumres=0;
rown=0.5;
trtseql=1;
trtseql='THS 2.2 Menthol - mCC';
output;
trtseql=2;
trtseql='mCC - THS 2.2 Menthol';
output;
trtseql=3;
trtseql='THS 2.2 Menthol - NRT Gum';
output;
trtseql=4;
trtseql='NRT Gum - THS 2.2 Menthol';
output;
trtseql=5;
trtseql='Enrolled not randomized';
output;
trtseql=99;
trtseql='Overall Safety';
output;
run;

data tot2;
merge dumtrts(in=a drop=dumres rown) tot(in=b);
if a or b;
if a and not b then count=0;
by trtseql;;
rename count=total;

```

```

        call symput('trt' || compress(put(trtseqan,best.)),
compress(count));
run;

proc transpose data=tot2 out=sum prefix=_;
    id trtseqan;
    var total;
run;

data sum01;
    set sum;
    total_overall=_1+_2+_3+_4+_5;
    bynum=1;
    rename _1=total_1 _2=total_2 _3=total_3 _4=total_4 _5=total_5;
    drop _name_ _label_;
run;

data table01;
    merge sum01 table;
    attrib      sub1 sub2 sub3 sub4 sub5 ov_sub label="n"
                p1 p2 p3 p4 p5 ov_p label='(%)' length=$8.
                ev1 ev2 ev3 ev4 ev5 ov_ev label="Events";

    by bynum;
    if sub1>0 then p1a=(sub1/total_1)*100;
    if sub2>0 then p2a=(sub2/total_2)*100;
    if sub3>0 then p3a=(sub3/total_3)*100;
    if sub4>0 then p4a=(sub4/total_4)*100;
    if sub5>0 then p5a=(sub5/total_5)*100;
    if ov_sub>0 then ov_pa=(ov_sub/total_overall)*100;

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

    if p4a=100 then p4=trim('(|compress(put(p4a,8.))||' %')');
    else if not missing(p4a) and p4a>10 and p4a<100 then p4=trim('(
||compress(put(p4a,8.1))||'%)');
    else if not missing(p4a) and p4a<10 then p4=trim('(
||compress(put(p4a,8.1))||'%)');

```

```

        if p5a=100 then p5=trim('('||compress(put(p5a,8.))||' %)');
        else if not missing(p5a) and p5a>10 and p5a<100 then p5=trim('(
||compress(put(p5a,8.1))||'%)');
        else if not missing(p5a) and p5a<10 then p5=trim('(
||compress(put(p5a,8.1))||'%)');

        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))||'%)');
        else if not missing(ov_pa) and ov_pa<10 then ov_p=trim('(
||compress(put(ov_pa,8.1))||'%)');/* end 7) aob 05mar2014 */

        array a [6] sub1 sub2 sub3 sub4 sub5 ov_sub;
        array b [6] ev1 ev2 ev3 ev4 ev5 ov_ev;
        do i=1 to 6;
            if a[i]=0 and b[i]=0 then b[i] =.;
        end;
        if missing(variable) then delete;
run;

proc sql noprint;
    create table table.T_15_02_06_18 as
    select variable, sub1, p1, ev1, sub2, p2, ev2, sub3, p3, ev3, sub4,
p4, ev4, sub5, p5, ev5, ov_sub, ov_p, ov_ev
    from table01
    order by rown;
quit;

data paging;
    set table01;
    by 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.)));
        flag=1;
run;

/* Standard - leave this */
options number nodate orientation=landscape papersize=&p_pgsize 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;
/* Standard - macro for paging */
%macro outrtf(blankn=, halfblnk=);

%if &halfblnk=N %then %let halfblnk=;
%else %if &halfblnk=Y %then %let halfblnk=~;

```

```

ods path stdlib.t106326 (read) ;
ods results off;
ods rtf toc_data
file="/cvn/projects/prj/data/000000106326/TFL/&TFL_Part./&tflno..rtf"
style=t106326 startpage=yes headery=1440 footery=1440 ;
ods noproctitle;
%do i=1 %to &page;

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

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

    /* Amend title as needed */
    _firtitl="Table 15.2.6.18 Summary of Cough Assessments -
Safety Population";
    _upcas=(length("Path: &TFLpath.")-
length(compress("Path:&TFLpath.",'ABCDEFGHIJKLMNOPQRSTUVWXYZ')))/2;
    len=&blankn.-ceil(_upcas)-length("Path: &TFLpath.")+6-length("(Page
&i of &page)");
    if eof then do;
        call symput('_FSRTITL', trim(left(_firtitl)));
        call symput('_blankn', compress(put(len,best.)));
        if page in (2 3 4) then call symput('page_x','1');
    end;
    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 */
ods listing close;
proc report data = comp style(header)={just=c} headline headskip nowd
split = '$' %if &i=1 %then %do; contents=' ' %end; %else %do; contents=' '
%end;;
    column flag page sect ("Variable" variable)
("Sequence &linebot" ("THS 2.2 Menthol -$mCC $(N=&trt1)"("n (%) Events"
sub1 p1 ev1))
("mCC -$THS 2.2 Menthol$(N=&trt2)"("n (%) Events" sub2 p2 ev2))
("THS 2.2 Menthol -$NRT gum $(N=&trt3)"("n (%) Events" sub3 p3 ev3))
("NRT gum -$THS 2.2 Menthol$(N=&trt4)"("n (%) Events" sub4 p4 ev4))

```

```

("Enrolled Not$Randomized$(N=&trt5) "("n (%) Events" sub5 p5 ev5)))
("Overall$Safety$(N=&trt99) "("n (%) Events" ov_sub ov_p ov_ev)); ;

        define flag                / order order = internal noprint;
        define page                / order order = internal noprint;
        define sect                / order order = internal noprint;
        define variable            / display style={just=l
cellwidth=3.4cm}"";
        define sub1-sub5          / display style={just=d
cellwidth=0.3cm}"";
        define ev1-ev5            / display style={just=c
cellwidth=0.4cm}"";
        define ov_sub             / display style={just=d
cellwidth=0.3cm}"";
        define ov_ev              / display style={just=c
cellwidth=0.4cm}"";
        define p1                 / display style={just=c
cellwidth=1.1cm}"";
        define p2                 / display style={just=c
cellwidth=1.1cm}"";
        define p3                 / display style={just=c
cellwidth=1.1cm}"";
        define p4                 / display style={just=c
cellwidth=1.1cm}"";
        define p5                 / display style={just=c
cellwidth=1.1cm}"";
        define ov_p               / display style={just=c cellwidth=1.1cm}"";

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

break after page / page;

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

compute after sect/style={just=left cellwidth=5cm
protectspecialchars=off};
    line "";
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: mCC = menthol conventional cigarettes; NRT gum =
Nicotine Replacement Therapy gum; THS = Tobacco Heating System.';
        line 'Note: Enrolled Not Randomized refers to all subjects
enrolled but not randomized. Overall Safety refers to all subjects
exposed to THS 2.2 Menthol or NRT gum.';
        line "Note: Percentages are based on the number of subjects
indicated in the column header (N).";
        line 'Note: Cough experienced in the previous 24h by the
subjects is assessed in the morning of Day 0 to Day 4 (to assess Day -1
to Day 3).';
        line "Note: If subject has answered question more than once
then the most severe intensity is presented.";
        line ' ';
        line 'Appendix 15.3.6.13';
        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=138, halfblnk=N);

ods listing;
proc printto print = "&table./T_15_02_06_18.lst" new;
run;

proc contents data = table.T_15_02_06_18 varnum;
run;
ods listing close;

proc printto ; run;
*=====;
* END OF PROGRAM CODE ;
*=====;

```