

```
/*=====
=====*
```

```
| Covance Study Number   : 000000106343          |
| Program Name           : t_conmed_atc.sas        |
| Purpose                 : Summary of Concomitant Medication by ATC          |
| Input Data              : ADAM.ADSL, ADAM.ADCM    |
| Output Data             : tlfds.T_15_02_06_19_01 & RTF          |
| Macros Called           : m_printto, m_logchk     |
| Originally Performed by : kpothuri                |
| Date                    : 01MAY2015               |
```

```
|
|=====
=====|
```

```
| Modification History          |
|-----|
| Modified by                   :                    |
| Modification Date             :                    |
| Modification Description      :                    |
```

```
+=====
=====*/
```

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

```
ods _all_ close;
```

```
ods listing;
```

```
%m_printto(route=YES);
```

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

```

* START OF PROGRAM CODE                                ;

*=====;

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

%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", ""));

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

run;

data cm;

    set adam.adcm;

    output;

    trta="Total";

    trtan=1;

    output;

run;

*N - counts - pre-randomization;

data adsl;

```

```

        set adam.adsl;

        output;

        trt01a="Total";

        trt01an=1;

        output;

run;

proc sql noprint;

        create table big_n as

        select trt01an, trt01a, count(distinct(usubjid))

        from adsl

        where SAFBFL="Y"

        group by trt01an, trt01a;

quit;

data big_n;

        set big_n;

        if trt01a="THSm2.2" then trt01a="THS";

        if trt01a="Product Test" then trt01a="ProdT";

run;

data _null_;

        set big_n;

        call symput ("B_" ||compress(trt01a), put(_TEMG001,8.));

run;

%put &B_SA &B_THS &B_mCC &B_ProdT &B_Total;

```

*N - counts - after randomization;

```
proc sql noprint;
```

```
    create table big_n_aft as
```

```
    select trt01an, trt01a, count(distinct(usubjid))
```

```
    from adsl
```

```
    where SAFAFL="Y"
```

```
    group by trt01an, trt01a;
```

```
quit;
```

```
data big_n_aft;
```

```
    set big_n_aft;
```

```
    if trt01a="THSm2.2" then trt01a="THS";
```

```
    if trt01a="Product Test" then trt01a="ProdT";
```

```
run;
```

```
data _null_;
```

```
    set big_n_aft;
```

```
    call symput ("A_" || compress(trt01a), put(_TEMG001,8.));
```

```
run;
```

```
%put &A_SA &A_THS &A_mCC &A_Total;
```

```
%macro pd (tab=, name=, where=, outdsn=);
```

```
*any medication;
```

```
proc sql noprint;
```

```
    create table anymed as
```

```
    select trta, trtan, count(distinct(usubjid))
```

```

from cm

where CMFL="Y" and &where

group by trta, trtan;

create table anymed_ev as

select trta, trtan, count(usubjid)

from cm

where CMFL="Y" and &where

group by trta, trtan;

quit;

proc transpose data=anymed out=t_anymed prefix=N_;

    id trta;

    var _temg001;

run;

proc transpose data=anymed_ev out=t_anymed_ev prefix=E_;

    id trta;

    var _temg001;

run;

data med;

length cmatc $200;

merge t_anymed t_anymed_ev;

by _name_;

```

```

ord=1;

cmtc="Any medication";

*pre-randomization;

%if &tab=1 %then %do;

    if not missing (N_SA) then np_SA=strip(put(N_SA,8.) || " (" ||
compress(put(round(((N_SA/&B_SA)*100),.01),5.1)) || ")");

    if not missing (N_THSM2_2) then np_THS=strip(put(N_THSM2_2,8.) || " (" ||
compress(put(round(((N_THSM2_2/&B_THS)*100),.01),5.1)) || ")");

    if not missing (N_MCC) then np_MCC=strip(put(N_MCC,8.) || " (" ||
compress(put(round(((N_MCC/&B_MCC)*100),.01),5.1)) || ")");

    if not missing (N_PRODUCT_TEST) then np_PRODT=strip(put(N_PRODUCT_TEST,8.) || " (" ||
compress(put(round(((N_PRODUCT_TEST/&B_ProdT)*100),.01),5.1)) || ")");

    np_TOTAL=strip(put(N_TOTAL,8.) || " (" ||
compress(put(round(((N_TOTAL/&B_Total)*100),.01),5.1)) || ")");

%end;

*after randomization;

%if &tab=2 %then %do;

    if not missing (N_SA) then np_SA=strip(put(N_SA,8.) || " (" ||
compress(put(round(((N_SA/&A_SA)*100),.01),5.1)) || ")");

    if not missing (N_THSM2_2) then np_THS=strip(put(N_THSM2_2,8.) || " (" ||
compress(put(round(((N_THSM2_2/&A_THS)*100),.01),5.1)) || ")");

    if not missing (N_MCC) then np_MCC=strip(put(N_MCC,8.) || " (" ||
compress(put(round(((N_MCC/&A_MCC)*100),.01),5.1)) || ")");

    if not missing (N_TOTAL) then np_TOTAL=strip(put(N_TOTAL,8.) || " (" ||
compress(put(round(((N_TOTAL/&A_Total)*100),.01),5.1)) || ")");

%end;

```

```
run;
```

```
data dum;
```

```
    ord=1.5;
```

```
    cmatc="";
```

```
    output;
```

```
run;
```

```
data med_1;
```

```
    merge med dum;
```

```
    by ord cmatc;
```

```
run;
```

```
*ATC1, ATC2;
```

```
proc sql noprint;
```

```
    create table atc1 as
```

```
    select trta, trtan, CMATC1, count(distinct(usubjid))
```

```
    from cm
```

```
    where CMFL="Y" and &where
```

```
    group by trta, trtan, CMATC1;
```

```
    create table atc2 as
```

```
    select trta, trtan, CMATC1, CMATC2, count(distinct(usubjid))
```

```
    from cm
```

```
    where CMFL="Y" and &where
```

```
group by trta, trtan, CMATC1, CMATC2;
```

```
create table atc1_ev as
```

```
select trta, trtan, CMATC1, count(usubjid)
```

```
from cm
```

```
where CMFL="Y" and &where
```

```
group by trta, trtan, CMATC1;
```

```
create table atc2_ev as
```

```
select trta, trtan, CMATC1, CMATC2, count(usubjid)
```

```
from cm
```

```
where CMFL="Y" and &where
```

```
group by trta, trtan, CMATC1, CMATC2;
```

```
quit;
```

```
proc sort data=atc1; by CMATC1; run;
```

```
proc transpose data=atc1 out=t_atc1 (drop=_name_) prefix=N_;
```

```
id trta;
```

```
var _temg001;
```

```
by CMATC1;
```

```
run;
```

```
proc sort data=atc2; by CMATC1 CMATC2; run;
```

```
proc transpose data=atc2 out=t_atc2 (drop=_name_) prefix=N_;
```

```
id trta;
```



```

        var _temg001;

        by CMATC1 CMATC2;

run;

proc sort data=atc1_ev; by CMATC1; run;

proc transpose data=atc1_ev out=t_atc1_ev (drop=_name_) prefix=E_;

    id trta;

    var _temg001;

    by CMATC1;

run;

proc sort data=atc2_ev; by CMATC1 CMATC2; run;

proc transpose data=atc2_ev out=t_atc2_ev (drop=_name_) prefix=E_;

    id trta;

    var _temg001;

    by CMATC1 CMATC2;

run;

*ATC1 merge;

data atc1_stat;

    merge t_atc1 t_atc1_ev;

    by CMATC1;

    *pre-randomization;

    %if &tab=1 %then %do;

```

```

        if not missing (N_SA) then np_SA=strip(put(N_SA,8.) || " (" ||
compress(put(round(((N_SA/&B_SA)*100),.01),5.1)) || ")");

        if not missing (N_THSM2_2) then np_THS=strip(put(N_THSM2_2,8.) || " (" ||
compress(put(round(((N_THSM2_2/&B_THS)*100),.01),5.1)) || ")");

        if not missing (N_MCC) then np_MCC=strip(put(N_MCC,8.) || " (" ||
compress(put(round(((N_MCC/&B_MCC)*100),.01),5.1)) || ")");

        if not missing (N_PRODUCT_TEST) then np_PRODT=strip(put(N_PRODUCT_TEST,8.) || " (" ||
compress(put(round(((N_PRODUCT_TEST/&B_ProdT)*100),.01),5.1)) || ")");

        np_TOTAL=strip(put(N_TOTAL,8.) || " (" ||
compress(put(round(((N_TOTAL/&B_Total)*100),.01),5.1)) || ")");

    %end;

*after randomization;

%if &tab=2 %then %do;

        if not missing (N_SA) then np_SA=strip(put(N_SA,8.) || " (" ||
compress(put(round(((N_SA/&A_SA)*100),.01),5.1)) || ")");

        if not missing (N_THSM2_2) then np_THS=strip(put(N_THSM2_2,8.) || " (" ||
compress(put(round(((N_THSM2_2/&A_THS)*100),.01),5.1)) || ")");

        if not missing (N_MCC) then np_MCC=strip(put(N_MCC,8.) || " (" ||
compress(put(round(((N_MCC/&A_MCC)*100),.01),5.1)) || ")");

        np_TOTAL=strip(put(N_TOTAL,8.) || " (" ||
compress(put(round(((N_TOTAL/&A_Total)*100),.01),5.1)) || ")");

    %end;

run;

proc sort data=atc1_stat out=atc1_stat_1; by descending n_total descending e_total; run;

data atc1_stat_1;

    set atc1_stat_1;

    ord=25;

```

```

        ord_atc1=_n_;

run;

data dummy;

        set atc1_stat_1;

        keep cmatc1 ord_atc1;

run;

*ATC2 merge;

data atc2_stat;

        merge t_atc2 t_atc2_ev;

        by CMATC1 CMATC2;


*pre-randomization;

%if &tab=1 %then %do;

        if not missing (N_SA) then np_SA=strip(put(N_SA,8.) || " (" ||
compress(put(round(((N_SA/&B_SA)*100),.01),5.1)) || ")");

        if not missing (N_THSM2_2) then np_THS=strip(put(N_THSM2_2,8.) || " (" ||
compress(put(round(((N_THSM2_2/&B_THS)*100),.01),5.1)) || ")");

        if not missing (N_MCC) then np_MCC=strip(put(N_MCC,8.) || " (" ||
compress(put(round(((N_MCC/&B_MCC)*100),.01),5.1)) || ")");

        if not missing (N_PRODUCT_TEST) then np_PRODT=strip(put(N_PRODUCT_TEST,8.) || " (" ||
compress(put(round(((N_PRODUCT_TEST/&B_ProdT)*100),.01),5.1)) || ")");

        np_TOTAL=strip(put(N_TOTAL,8.) || " (" ||
compress(put(round(((N_TOTAL/&B_Total)*100),.01),5.1)) || ")");

%end;

```

```

*after randomization;

%if &tab=2 %then %do;

    if not missing (N_SA) then np_SA=strip(put(N_SA,8.) || " (" ||
compress(put(round(((N_SA/&A_SA)*100),.01),5.1)) || ")");

    if not missing (N_THSM2_2) then np_THS=strip(put(N_THSM2_2,8.) || " (" ||
compress(put(round(((N_THSM2_2/&A_THS)*100),.01),5.1)) || ")");

    if not missing (N_MCC) then np_MCC=strip(put(N_MCC,8.) || " (" ||
compress(put(round(((N_MCC/&A_MCC)*100),.01),5.1)) || ")");

    np_TOTAL=strip(put(N_TOTAL,8.) || " (" ||
compress(put(round(((N_TOTAL/&A_Total)*100),.01),5.1)) || ")");

    %end;

run;

proc sort data=dummy; by cmatc1; run;

proc sort data=atc2_stat; by cmatc1; run;

data atc2_ord_1;

    merge dummy atc2_stat;

    by cmatc1;

    ord=50;

run;

proc sort data=atc2_ord_1; by descending n_total cmatc2; run;

data atc2_ord_2;

    set atc2_ord_1;

    ord_atc2=_n_;

run;

```

```
*combine atc1 and atc2 sets;
```

```
data subset;
```

```
    set atc1_stat_1 atc2_ord_2;
```

```
run;
```

```
proc sort data=subset; by cmatc1; run;
```

```
data subset1;
```

```
length cmatc2_ $200;
```

```
    set subset;
```

```
    cmatc2_ = " " || cmatc2;
```

```
    if cmatc2_ = "" then cmatc2_=cmatc1;
```

```
run;
```

```
proc sort data=subset1 out=sub_ord; by ord_atc1 ord_atc2; run;
```

```
data comb;
```

```
    set med_1(rename=(cmatc=cmatc2_)) sub_ord;
```

```
    cmatc=propcase(cmatc2_);
```

```
    column1=strip(cmatc);
```

```
    if ord_atc1=. then ord_atc1=0;
```

```
run;
```

```
data &outdsn;
```

```
    set comb;
```

```
    attrib column column1 wrap format=$400.;
```

```

        tabnum=&tab;

        set_name="&name";

wrap = column1;

if ord=50 then do;

i=50; *this is the max length allowed on a single line - change as needed;

if length(wrap)>i then do;

    nwraps = int(length(wrap)/i); *calculate how many lines the text will wrap over;

    do while(nwraps > 0);

        fin=0;

        j = i*nwraps; *calculate starting point - loop will cycle backwards from this point looking for a space;

            test=j;

        do while(fin=0 AND J GT 1);

            if substr(wrap,j,1)=' ' 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;

column="|S={foreground=white} . |S={} " || wrap;

end;

```

```

else column=column1;

run;

%mend;

%pd (tab=1, name=prior rand, where=%str(SAFBFL="Y" and ASPER=1), outdsn=b_rand);

%pd (tab=2, name=after rand, where=%str(SAFAFL="Y" and ASPER in (2,3,4)), outdsn=a_rand);


data comb (drop=_: n_);

    set b_rand a_rand;


    if ord ^= 1.5 then do;

        if np_SA = "" then np_SA = "0";

        if np_THS = "" then np_THS = "0";

        if np_MCC = "" then np_MCC = "0";

        if np_PRODT = "" then np_PRODT = "0";

        if np_TOTAL = "" then np_TOTAL = "0";


        if e_SA = . then e_SA = 0;

        if e_THSm2_2 = . then e_THSm2_2 = 0;

        if e_MCC = . then e_MCC = 0;

        if e_PRODUCT_TEST = . then e_PRODUCT_TEST = 0;

        if e_TOTAL = . then e_TOTAL = 0;

    end;

    else do;

        call missing(np_SA, np_THS, np_MCC, np_PRODT, np_TOTAL, e_SA, e_THSm2_2,
e_MCC, e_PRODUCT_TEST, e_TOTAL);

```

end;

if column1="Antidiarrheals, Intestinal Antiinflammatory/Antiinfective Agents" then

column="|S={foreground=white} . |S={} Antidiarrheals,|n |S={foreground=white} . |S={} Intestinal|n |S={foreground=white} . |S={} Antiinflammatory/Antiinfective|n |S={foreground=white} . |S={} Agents";

*line number may need update;

if tabnum=1 then line=_n_;

else if tabnum=2 then line=_n_-39;

run;

/*Check for page overflows, this may need changing*/

data final;

set comb;

if tabnum=1 then do;

if ln gt 4 then ln=1;

else ln+1;

if ln=1 then page+1;

end;

else if tabnum=2 then do;

if ln_ gt 4 then ln_=1;

else ln_+1;

if ln_=1 then page_+1;

end;

run;


```
data final_1 (drop=page rename=(page_=page));  
    set final;  
  
    if page_=0 then page_=page;  
    if tabnum=2 then do;  
        page_=page_+8; *may need to change;  
    end;  
  
    call symput("page",compress(put(page_,best.)));  
run;  
%put &page;
```

```
%let tflno = %str(T_15_02_06_19_01);
```

```
data tflds.&tflno;  
    set final_1;  
run;
```

```
proc sort data=tflds.&tflno out=final_2; by tabnum ord_atc1 cmatc1 ord_atc2; run;
```

```
data final_3;  
    set final_2;  
    where ord=25;  
    if NP_SA="0" then flag_sa=1;  
    if NP_THS="0" then flag_ths=1;
```

```

if NP_MCC="0" then flag_mcc=1;

if NP_PRODT="0" then flag_pt=1;


if E_SA=0 then flag_sa_e=1;

if E_THSM2_2=0 then flag_ths_e=1;

if E_MCC=0 then flag_mcc_e=1;

if E_PRODUCT_TEST=0 then flag_pt_e=1;


keep flag_: ord_atc1 tabnum;

run;


data comb_fin;

merge final_2 final_3;

by tabnum ord_atc1;


if ord=50 then do;

    if flag_ths=1 then NP_THS=" ";

    if flag_mcc=1 then NP_MCC=" ";

    if flag_sa=1 then NP_SA=" ";

    if flag_pt=1 then NP_PRODT=" ";


    if flag_sa_e=1 then E_SA=.;

    if flag_ths_e=1 then E_THSM2_2=.;

    if flag_mcc_e=1 then E_MCC=.;

    if flag_pt_e=1 then E_PRODUCT_TEST=.;

```

```

        end;

run;

/* Standard - leave this */

options number nodate orientation=landscape /* papersize=&P_PGSIZE*/ missing=' ';

%let linetop = \brdrt\brdrs\brdrw30; * needs to be 1.5pt so calculated in twips (1/20 pt) ;

%let linebot = \brdrb\brdrs\brdrw30;

ods escapechar='|';

/* Standard - macro for paging */

%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 file="/cvn/projects/prj/data/000000106343/TFL/&TFL_Part./Tables/&tflno..rtf"
style=t106343 startpage=yes headery=1440 footery=1440 ;

ods noproctitle;

%do i=1 %to &page;

title ;

```

```

footnote;

%let wd=0;

%LET NOOBS=0;

ods proclabel = ' ';


data comp;

    set comb_fin end=eof;

        where page=&i;

            /* Amend title as needed */

                _firtitl="Table 15.2.6.19.1 Summary of Concomitant Medication by Anatomical
Therapeutic Classes (ATC) 1 and 2 - Safety";

                _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('period1', compress(put(tabnum,best.)));

                        call symput('_blankn', compress(put(len,best.)));

                    end;

                drop _firtitl _upcas len;

run;


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 nowd split = '$' %if &i=1 %then %do; contents=' ' %end;  
%else %do; contents=" %end;;;
```

```
column page line column
```

```
%if &period1.=1 %then %do; ("THSm2.2$(N=%left(&B_THS))&linebot" np_ths e_thsm2_2)  
("mCC$(N=%left(&B_mCC))&linebot" np_mcc e_mcc)
```

```
("SA$(N=%left(&B_SA))&linebot" np_sa e_sa) ("Product Test$(N=%left(&B_Prodt))&linebot" np_prodt  
e_product_test)
```

```
("Overall Safety$(N=%left(&B_Total))&linebot" np_total e_total); %end;
```

```
%if &period1.=2 %then %do; ("THSm2.2$(N=%left(&A_THS))&linebot" np_ths e_thsm2_2)  
("mCC$(N=%left(&A_mCC))&linebot" np_mcc e_mcc)
```

```
("SA$(N=%left(&A_SA))&linebot" np_sa e_sa)
```

```
("Overall Safety$(N=%left(&A_Total))&linebot" np_total e_total); %end;
```

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

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

```
define column /"ATC1$""~\~ATC2' display style(column)={just=left cellwidth=4.2cm  
asis=on} style(header)={just=left} "";
```

```
define np_ths /"n (%)" display style={just=c cellwidth=1.4cm}  
style(header)={just=center} ;
```

```
define e_thsm2_2 /"Events" display style={just=c cellwidth=1.0cm}  
style(header)={just=center};
```

```
define np_mcc /"n (%)" display style={just=c cellwidth=1.2cm}  
style(header)={just=center} ;
```

```

define e_mcc      /"Events" display style={just=c cellwidth=1.0cm}
style(header)={just=center};

define np_sa      /"n (%)" display style={just=c cellwidth=1.2cm}
style(header)={just=center} ;

define e_sa      /"Events" display style={just=c cellwidth=1.0cm}
style(header)={just=center};

%if &period1.=1 %then %do;

define np_prodt   /"n (%)" display style={just=c cellwidth=1.4cm}
style(header)={just=center} ;

define e_product_test /"Events" display style={just=c cellwidth=1.0cm}
style(header)={just=center};

%end;

define np_total   /"n (%)" display style={just=c cellwidth=1.1cm}
style(header)={just=center} ;

define e_total    /"Events" display style={just=c cellwidth=1.0cm}
style(header)={just=center};

```

break after page / page;

compute after page;

line " ";

endcomp;

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

line "&linetop";

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 "Population";

line "";

%if &period1.=1 %then %do; line "Safety Time Period: Pre-Randomization"; %end;

%else %do; line "Safety Time Period: Randomization"; %end;

line "&linebot";

endcomp;

compute after _page_ / style={just=left protectspecialchars=off pretext="&linetop."};

%if &period1.=1 %then %do;

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.';

%end;

%else %do; line 'Note: The Overall Safety refers to all subjects exposed to THSm2.2.';
%end;

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 15.3.6.4';

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=50, halfblnk=N);
```

```
ods listing;
```

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

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

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

```
%m_logchk;
```