

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

```
*Covance Study ID   : 000000106343
```

```
*Program Name       : t_co_fas.sas
```

```
*Purpose            : Descriptive Statistics of Exhaled CO (ppm) - FAS
```

Table 15.2.4.6.2

```
*Input Data         : adam.adsl, ADAM.adbx
```

```
*Output Data        : tflds.T_15_02_04_06_02, tflds.T_15_02_04_06_02_F
```

```
*Macros Called       : %m_printto, %m_logchk, %mmeans, %outrtf
```

```
*Programmed by      : L.Ma
```

```
*Creation Date       : 2015-05-15
```

```
*=====
```

```
*Modification History
```

```
*Date    Initials  No. Reason;
```

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

```
options notes nosource;
```

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

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

```
ods _all_ close;
```

```
ods listing;
```

```
%m_printto;
```

```
/*Use ADSL to get N values for column headers*/
```

```

data adsl;

    set adam.adsl(where=(fasfl='Y'));

    if          trt01pn=4 then trt=1;

    else if trt01pn=5 then trt=2;

    else if trt01pn=3 then trt=3;

run;

```

```

proc freq data=adsl noprint;

    table trt/ out=tot(drop=percent rename=(count=total));

run;

```

```

data tot2;

    set tot;

    call symput('trt' || compress(put(trt,best.)), compress(put(total, best.)));

run;

```

/\*Bring in data from ADBX for Exhaled CO - FAS per Mock\*/

```

data adbx1;

    set adam.adbx;

    where anl02fl='Y' AND PARCAT1 = 'BIOMARKERS' AND PARAMCD = 'CO' AND LBSPEC = 'EXPIRED
AIR'

        AND 190>= AVISITN >=100

        AND (fasfl='Y');

    if          trtpn=4 then trt=1;

    else if trtpn=5 then trt=2;

```

```

else if trtpn=3 then trt=3;

run;

data adbx;

    set adbx1;

    if ablfl='Y' then avisit='Baseline';

    *According to 5/8/2015 4:04 PM (and 6/3/2015)email from John;

    *for avisit=Baseline and atpt=DAY 0 - WITHIN 15 MIN PRIOR TO SMOKING and avisit=Baseline
and atpt=DAY 1 - WITHIN 15 MIN PRIOR TO SMOKING

    should combined for one by basetype number part =(1). also need to change the atptn and
avisitn for the order purpose on the table per mock;

    if avisit="Baseline" and atpt="DAY 0 - WITHIN 15 MIN PRIOR TO SMOKING" and trt=3 then do;
atpt="DAY 1 - 08:00 - 09:30"; atptn=1.11; avisitn=100; end;

    if avisit="Baseline" and atpt="DAY 0 - WITHIN 15 MIN PRIOR TO SMOKING" then do;
atpt="BASELINE - WITHIN 15 MIN PRIOR TO SMOKING"; atptn=1.1; avisitn=100; end;

    if avisit="Baseline" and atpt="DAY 1 - WITHIN 15 MIN PRIOR TO SMOKING" then do;
atpt="BASELINE - WITHIN 15 MIN PRIOR TO SMOKING"; atptn=1.1; avisitn=100; end;

    if avisit="Baseline" and atpt="DAY 1 - 08:00 - 09:30" then do; atpt="BASELINE - 08:00 - 09:30";
atptn=1.11; avisitn=100; end;

    if avisit="Day 1" and atpt="DAY 1 - WITHIN 15 MIN PRIOR TO SMOKING" then delete;

    if aval ne 0 and aval ne . then logaval=log(aval);

    *if avisit not in (Baseline DAY 1 to DAY 90) then delete;

    if avisit='Day 0' then delete;

    **** Per JH email on 5-18-2015 8:58am: following subjects with avisit=Day 1 and atpt=DAY 1 -
08:00-09:30 should not need to be presented on stat summary. ****;

```

```

        if avisit="Day 1" and atpt="DAY 1 - 08:00 - 09:30" then delete;

run;

*****
;

* macro for general mean stats(n mean std median min max Q25 Q75 lclm uclm) per mock ;

*****
;

%macro mmeans(dsn=, class=, var=, out=);

proc means data=&dsn. noprint nway;

    var &var.;

    class &class. trt;

    output out=results02 n=n1 mean=mean1 std=std1 median=median1 min=min1 max=max1 q1=q1
q3=q3 lclm=lci1 uclm=uci1;

run;


data results03;

    set results02;

    attrib meansd length=$20.

        minmax length=$20.

        n    length=$20.

        miss length=$20.

        median length=$20.

        quart aci length=$20.;


n = left(compress(put(n1,8.)));

```

```

*for <missing, n(%)> row;

if trt=1 then do;

                                *format missing cell to 0 per John email on 8-5-2015;

                                if &trt1.=n1 then miss="0";

                                else miss=strip(put((&trt1.-n1), 8.)) || ' ' || strip(put(((&trt1.-
n1)*100)/&trt1., 8.1)) || "");

                                end;

else if trt=2 then do;

                                if &trt2.=n1 then miss="0";

                                else miss=strip(put((&trt2.-n1), 8.)) || ' ' || strip(put(((&trt2.-
n1)*100)/&trt2., 8.1)) || "");

                                end;

else if trt=3 then do;

                                if &trt3.=n1 then miss="0";

                                else miss=strip(put((&trt3.-n1), 8.)) || ' ' || strip(put(((&trt3.-
n1)*100)/&trt3., 8.1)) || "");

                                end;

if not missing(median1) then median = left(compress(put(round(median1,0.01),8.2)));

if not missing(mean1) and not missing(std1) then meansd =
left(compress(put(round(mean1,0.01),8.2))) || ' (' || left(compress(put(0.001*ceil(std1/0.001),8.3))) ||
');

if not missing(min1) and not missing(max1) then minmax = left(compress(put(min1,8.1))) || ', ' ||
left(compress(put(max1,8.1)));

if not missing(lci1) and not missing(uci1) then aci = strip(put(0.01*floor(lci1/0.01),8.2)) || ', ' ||
strip(put(0.01*ceil(uci1/0.01),8.2));

if not missing(q1) and not missing(q3) then quart = strip(strip(put(round(q1, 0.01),8.2)) || ', ' ||
strip(put(round(q3, 0.01),8.2)));

```

```

drop n1 mean1 std1 median1 min1 max1 q1 q3 uci1 lci1 ;

run;

proc transpose data=results03 out=&out prefix=r name=varname;

    by &class.;

    var n miss meansd median minmax aci quart;

    id trt;

run;

data &out.;

    set &out.;

    length stat $200;

    if upcase(varname)='N'          then do; statord=1; stat='n'; end;

    if upcase(varname)='MISS'       then do; statord=2; stat='Missing, n (%)'; end;

    if upcase(varname)='MEDIAN' then do; statord=7; stat='Median'; end;

    if upcase(varname)='QUART'     then do; statord=8; stat='Q25, Q75'; end;

    if upcase(varname)='MINMAX' then do; statord=9; stat='Min, Max'; end;

    if upcase(varname)='MEANSD' then do; statord=10; stat='Mean (SD)'; end;

    if upcase(varname)='ACI'       then do; statord=11; stat='95% CI of Mean'; end;

run;

%mend mmeans;

%mmeans(dsn=adbx, class=avisitn avisit atptn atpt, var=aval, out=out1);

%mmeans(dsn=adbx, class=avisitn avisit atptn atpt, var=pchg, out=out2);

```

\*\*\*\*\*

;

\* Geometric Mean per mock ;

\*\*\*\*\*

;

proc means data=adbx noprint nway;

var logaval;

class avisitn avisit atptn atpt trt;

output out=results02 mean=mean std=std1 lclm=lci1 uclm=uci1;

run;

\*as per email from John dated on Tuesday, May 19, 2015 11:02 AM:

CO and Ames data from ADBX sometimes have AVAL=0 – in those cases, please report the Geometric Mean

and 95% CI as NC if for a given treatment group at least one value is AVAL=0 (because you cannot take the log of 0).

Add a footnote to table as noted below from 04 example.;

proc freq data=adbx(where=(aval=0));

tables trt\*avisitn\*avisit\*atpt\*atptn\*aval/noprint out=aval0(drop=percent count aval) ;

run;

data fl0;

set aval0;

flg0='Y';

run;

```

proc sort data=f10;

    by avisitn avisit atptn atpt trt;

run;

data res0;

    merge results02(in=a) f10;

    by avisitn avisit atptn atpt trt;

    if a;

run;

data results03;

    set res0;

    gmean1=exp(mean);

    gmean=left(compress(put(round(gmean1,0.01), 8.2)));

    gcv=compress(put(0.001*ceil((sqrt(exp(std1*std1)-1)*100)/0.001),8.3));

    glci=exp(lci1);

    guci=exp(uci1);

    if not missing(gcv) then gmeancv=left(trim(gmean)) || ' (' || left(trim(gcv)) || ')'; else
gmeancv=left(trim(gmean));

    if not missing(glci) and not missing(guci) then ci = strip(strip(put(0.01*floor(glci/0.01),8.2)) || ', '
|| strip(put(0.01*ceil(guci/0.01),8.2)));

    if flg0='Y' then do; gmeancv='NC'; ci='NC'; end;

run;

```



```
proc transpose data=results03 out=out prefix=r name=varname;
```

```
by avisitn avisit atptn atpt;
```

```
var gmeancv ci;
```

```
id trt;
```

```
run;
```

```
data out3;
```

```
set out;
```

```
length stat $200;
```

```
if upcase(varname)='GMEANCV' then do; statord=5; stat='Geometric Mean (CV%)'; end;
```

```
if upcase(varname)='CI' then do; statord=6; stat='95% CI of Geometric Mean'; end;
```

```
run;
```

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

```
* BLOQ/ALOQ ;
```

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

```
*per Jh email at Wed 7/1/2015 9:34 AM ---- we only need to present BLOQ as a line item if there are  
BLOQ values for a given parameter/timepoint;
```

```
proc freq data=adam.adbx(where=(anl02fl='Y' AND PARCAT1 = 'BIOMARKERS' AND PARAMCD = 'CO'  
AND LBSPEC = 'EXPIRED AIR' AND AQLFL='Y' ));
```

```
table AVALC;
```

```
run;
```

```
*no data for BLOQ values so no need to present BLOQ n(%) row.;
```

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

```

* set together ;

*****.

*value part;

data final1;

    set out1 out3 ;

run;


proc sort data=final1 out=final2;

    by avisitn avisit atptn atpt statord stat varname;

run;


*change part;

proc sort data=out2 out=out2_s;

    by avisitn avisit atptn atpt statord stat varname;

run;


data outc;

    set out2_s;

    by avisitn avisit atptn atpt statord stat varname;

    if avisit='Baseline' then delete;

run;


*merge value part and change part to get all together;

data final;

    retain avisitn avisit atptn statord tp stat r1 c1 r2 c2 r3 c3 statord;

```

```

merge final2(in=a) outc(in=b rename=(r1=c1 r2=c2 r3=c3 ));

by avisitn avisit atptn atpt statord stat varname;

if a;

*format avisit/tp per mock; /*per JH email on Tue 4/28/2015 10:50 AM*/

tp=atpt;

if substr(tp,6,3)=' - ' then substr(tp,6,3)= ' , ' ;

if avisit="Baseline" then do;

    if index(tp,'DAY 0,') then tp='BASELINE, ' || tranwrd(tp,'DAY 0,','');

    else if index(tp,'BASELINE - WITHIN 15 MIN PRIOR TO SMOKING') then
tp=tranwrd(tp,'BASELINE - WITHIN 15 MIN PRIOR TO SMOKING','Baseline, Within 15 Min Prior To
Smoking'); /*per john email on Mon 7/6/2015 11:01 AM*/

    else if index(tp,'BASELINE -') then tp=tranwrd(tp,'BASELINE -','BASELINE, ');

end;

else do;

    if index(tp,'WITHIN 15 MIN PRIOR TO SMOKING') then tp=tranwrd(tp,'WITHIN 15 MIN
PRIOR TO SMOKING',' Within 15 Min Prior To Smoking'); /*per john email on Mon 7/6/2015 11:01
AM*/

end;

if avisit="Day 6/Discharge Confinement" then tp="DAY 6, DISCHARGE CONFINEMENT";

***delete <missing, n(%)> if no missing n data for the row;

if stat="Missing, n (%)" and r1="0" and r2="0" and r3="0" and c1="0" and c2="0" and c3="0"
then delete;

else if stat="Missing, n (%)" and avisit="Baseline" and r1="0" and r2="0" and r3="0" then
delete;

else if stat="Missing, n (%)" and avisit="Baseline" and index(tp,"Within 15 Min Prior To
Smoking")>0 and r1="0" and r2="0" and r3="" then delete;

```

```
        else if stat="Missing, n (%)" and avisit="Baseline" and index(tp,"08:00 - 09:30")>0 and r1="" and  
r2="" and r3="0" then delete;
```

```
        else if stat="Missing, n (%)" and index(tp,"Within 15 Min Prior To Smoking")>0 and r1="0" and  
c1="0" and r2="0" and c2="0" and r3="" and c3="" then delete;
```

```
        else if stat="Missing, n (%)" and index(tp,"08:00 - 09:30")>0 and r1="" and c1="" and r2="" and  
c2="" and r3="0" and c3="0" then delete;
```

```
        tp=propcase(tp);
```

```
        keep avisitn avisit atptn atpt statord tp stat r1 c1 r2 c2 r3 c3;
```

```
run;
```

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

```
*create new page for each avisit for report ;
```

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

```
proc sql;
```

```
    create table page as
```

```
    select distinct avisitn, atptn, tp
```

```
    from final
```

```
    order by avisitn, atptn, tp;
```

```
quit;
```

```
data page1;
```

```
    set page;
```

```
    by avisitn atptn tp;
```

```
    if _n_ = 0 then page = 0;
```

```
    page + 1;
```

```
run;
```

```

proc sql;

    create table final_page as

    select distinct a.*, b.page

    from final as a

    left join page1 as b

    on a.avisitn=b.avisitn and a.tp = b.tp and a.atptn=b.atptn

    order by page, avisitn, avisit, atptn, statord;

quit;


data final_page(rename=(r1=THSm c1=THSm_chg r2=mCC c2=mCC_chg r3=SA c3=SA_chg));

    set final_page end=last;


    if last then call symputx("page", page);

run;


/*output report data; */

%let tflno=T_15_02_04_06_02;

data tfls.&tflno(keep=avisitn avisit tp stat THSm mCC SA THSm_chg mCC_chg SA_chg statord page);

    set final_page;

run;


*****.

*create output report ;

```

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

```
options number nodate orientation=landscape 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=130, halfblnk=N);
```

```
%let title1 = %str(Table 15.2.4.6.2 Descriptive Statistics of Exhaled CO (ppm) - FAS);
```

```
%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", ""));
```

```
        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=1440 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;

        /* Amend title as needed */

        _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 tp stat ("THSm2.2$(N=&trt1)&linebot" THSm THSm_chg )
```

```
("mCC$(N=&trt2)&linebot" mCC mCC_chg)
```

```
("SA$(N=&trt3)&linebot" SA SA_chg);
```

```
define tp          /"Timepoint" order order=internal style={just=left cellwidth=0.9cm}  
style(header)={just=left} ;
```

```
define stat        /"Statistic" display style={just=left cellwidth=1.9cm}  
style(header)={just=left} ;
```

```
define THSm                /"Value" display style={just=c cellwidth=1.2cm}  
style(header)={just=center} ;
```

```
define mCC                /"Value" display style={just=c cellwidth=1.2cm}  
style(header)={just=center} ;
```

```
define SA                /"Value" display style={just=c cellwidth=1.2cm}  
style(header)={just=center};
```

```
define THSm_chg          /"% Change(*)" display style={JUST=c cellwidth=1cm}  
style(header)={just=center};
```

```
define mCC_chg          /"% Change(*)" display style={just=c cellwidth=1cm}  
style(header)={just=center};
```

```
define SA_chg          /"% Change(*)" display style={just=c cellwidth=1cm}  
style(header)={just=center};
```

```
compute after tp;
```

```
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; SA = Smoking abstinence;
THSm2.2 = Tobacco Heating System 2.2 Menthol.';

LINE 'Note: * % Change from baseline, where baseline is defined as the last
assessment prior to first randomized product use in mCC / THSm2.2 arms or the last assessment prior to
10AM on Day 1 in the SA arm.';

line 'Note: For patients randomized to SA, if the 08:00 - 09:30 Day 1
measurement was missing then the latest non-missing measurement within 15 minutes prior to smoking
on Day -1 or Day 0 was used.' ;

line 'Note: NC = Not Calculated.';

line ' ';

line 'Appendix 15.3.3.2';

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, halfblk=N);
```

```
ods listing;
```

```
%m_logchk;
```

```
/****** END OF FILE t_co_fas.sas *****/
```