

```

/*=====
| Covance Study Number      : 000000106343      |
| Program Name              : t_mCEQ_FAS.sas      |
| Purpose                   : Program to create table 15.2.4.54.2      |
| Input Data                : ADAM.ADSL, ADAM.ADQSPA      |
| Output Data              : T_15_02_04_54_02      |
| Macros Called             :                      |
| Originally Performed by   : Upender S          |
| Date                     : 28APR2015          |
|=====
| Modification History
|-----
| Modified by              :                      |
| Modification Date        :                      |
| Modification Description  :                      |
+=====*/

options spool replace;

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

%m_printto;

%let tflno=T_15_02_04_54_02;
%let title1 = Table 15.2.4.54.2 Descriptive Statistics of MCEQ subscales - FAS;

proc sql;

select count(distinct usubjid) into: NTHS from adam.adsl(where=(trt01pn = 4 and FASFL = "Y"));
select count(distinct usubjid) into: NMCC from adam.adsl(where=(trt01pn = 5 and FASFL = "Y"));
select count(distinct usubjid) into: NSAA from adam.adsl(where=(trt01pn = 3 and FASFL = "Y"));

quit;

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

data adqspa_1 (where=(FASFL='Y'));
set adam.adqspa;
    where ( 101<=avisitn <= 105 or avisitn = 130 or avisitn = 160 or avisitn = 190);
if strip(paramcd) in ("MCEQA" "MCEQCR" "MCEQERTS" "MCEQPR" "MCEQSS") and anl01f1 = "Y";
run;

data adqspa_2;
set adam.adqspa;
if strip(paramcd) in ("MCEQA" "MCEQCR" "MCEQERTS" "MCEQPR" "MCEQSS") and anl01f1 = "Y" and trtpn ne 3;
if ablfl = "Y" and FASFL = "Y" then do;
    avisitn = 10;
    avisit = "Baseline";
    apuper = 1;
    apuperc = "Period 1";
output;
end;
run;

data adqspa;
set adqspa_1 adqspa_2;
run;

proc sort data=adqspa;
by trtpn paramn param apuper apuperc avisitn avisit ;
run;

data n;

```

```

length txt $200.;
set adqspa;
if aval = . then do;
txtn = 2;
txt = "<Missing, n(%)>";
output;
end;
if index(avalc, "<") > 0 then do;
txtn = 3;
txt = "<BLOQ, n(%)>";
output;
end;
if index(avalc, ">") > 0 then do;
txtn = 4;
txt = "<ALOQ, n(%)>";
output;
end;
run;

proc sort data=n out=n1 nodupkey dupout=dup;
by usubjid paramn avisitn txtn ;
run;

proc freq data=n1 noprint;
tables paramn*param*apuper*apuperc*avisitn*avisit*txtn*txt*trtpn/out=n_freq;
run;

data n_freq;
length countx $50.;
set n_freq;
countx = strip(put(count, best.));
run;

proc sort data=n_freq;
by paramn param apuper apuperc avisitn avisit txtn txt;
run;

proc transpose data=n_freq out=n_freq_t prefix=trt_;
by paramn param apuper apuperc avisitn avisit txtn txt;
var countx;
id trtpn;
run;

data n_freq_t;
set n_freq_t;
where paramn ne .;
run;

proc sort data=adqspa;
by trtpn paramn param apuper apuperc avisitn avisit ;
run;

proc means data=adqspa(where=(aval ne .)) noprint;
var aval;
by trtpn paramn param apuper apuperc avisitn avisit ;
output out=aval n =n mean = mean std = std median = median min = min max = max q1 = q1 q3 = q3 lclm = lclm uclm = uclm;
run;

data aval1;
set aval;
length median1 Q2575 Minmax Meansd CIAM n1 $50.;
if not missing(median) then median1 = left(compress(put(round(median,0.1),8.1)));
if not missing(mean) and not missing(std) then meansd = left(compress(put(round(mean,0.1),8.1))) || ' (' || left(compress(put(0.01*ceil(std/0.01),8.2))) || ')';
if not missing(min) and not missing(max) then minmax = left(compress(put(min,8.))) || ', ' || left(compress(put(max,8.)));
if not missing(lclm) and not missing(uclm) then ciam = strip(put(0.1*floor(lclm/0.1),8.1)) || ', ' || strip(put(0.1*ceil(uclm/0.1),8.1));
if not missing(q1) and not missing(q3) then q2575 = strip(put(round(q1,0.1),8.1)) || ', ' || strip(put(round(q3,0.1),8.1));

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

if trtpn= 4 then BigN= &NTHS;
if trtpn= 5 then BigN = &NMCC;

If ^missing(BigN) and ^missing(n) then msng=BigN-n;
if .<msng^=0 then msng_prcnt=(msng/BigN)*100;

```

```
if .<msgng^=0 then msg_pct=compress(put(msgng, best.))||' ('||compress(put(msgng_prcnt, 5.1))||')';
```

```
run;
```

```
proc sort data=aval1;  
by paramn param apuper apuperc avisitn avisit ;  
run;  
data tflds.&tflno._f;  
set aval1;  
run;  
proc transpose data=aval1 out=aval_t prefix= trt_;  
by paramn param apuper apuperc avisitn avisit ;  
var n1 median1 q2575 minmax meansd ciam msg_pct;  
id trtpn;  
run;
```

```
data aval_t;  
length txt $200.;  
set aval_t;  
if upcase(_name_) = "N1" then do;  
txtn = 1;  
txt = "n";  
end;  
if upcase(_name_) = "MSG_PCT" then do;  
txtn = 1.5;  
txt = "Missing, n (%)";  
end;  
  
else if upcase(_name_) = "MEDIAN1" then do;  
txtn = 7;  
txt = "Median";  
end;  
else if upcase(_name_) = "Q2575" then do;  
txtn = 8;  
txt = "Q25, Q75";  
end;  
else if upcase(_name_) = "MINMAX" then do;  
txtn = 9;  
txt = "Min, Max";  
end;  
else if upcase(_name_) = "MEANSD" then do;  
txtn = 10;  
txt = "Mean (SD)";  
end;  
else if upcase(_name_) = "CIAM" then do;  
txtn = 11;  
txt = "95% CI of Mean";  
end;
```

```
run;
```

```
data adqspa_log;  
set adqspa;  
if aval ne . then logaval = log(aval);  
run;
```

```
proc sort data=adqspa_log;  
by trtpn paramn param apuper apuperc avisitn avisit ;  
run;
```

```
proc means data=adqspa_log noprint;  
by trtpn paramn param apuper apuperc avisitn avisit ;  
output out=aval_log mean = mean std = std lclm = lclm uclm = uclm;  
var logaval;  
run;
```

```
data aval_log1;  
set aval_log;  
length geocv CIGM $50.;  
if ^missing(mean) then gmean1=exp(mean);  
if ^missing(lclm) then glci=exp(lclm);  
if ^missing(uclm) then guci=exp(uclm);  
gmean=left(compress(put(round(gmean1,0.1), 8.1)));
```

```

    if ^missing(std) then gcv=compress(put(0.01*ceil((sqrt(exp(std*std)-1)*100)/0.01),8.2));
    if not missing(gcv) then geocv=left(trim(gmean)) || ' (' || left(trim(gcv))||')';
    else geocv=left(trim(gmean));
    if not missing(glci) and not missing(guci) then cigm = strip(strip(put(0.1*floor(glci/0.1),8.1)) || ', ' || strip(put(0.1*ceil(guci
/0.1),8.1)));

else if glci ne . and guci = . then cigm = strip(put(0.1*floor(glci/0.1), 8.1))||", NA";
else if glci = . and guci ne . then cigm = "NA, "||strip(put(0.1*ceil(guci/0.1),8.1));
else if glci = . and guci = . then cigm = "NA, NA";

run;

proc sort data=aval_log1 ;
by paramn param apuper apuperc avisitn avisit ;
run;

proc transpose data=aval_log1 out=aval_log1_t prefix= trt_;
by paramn param apuper apuperc avisitn avisit ;
var geocv cigm;
id trtpn;
run;

data aval_log1_t;
length txt $200.;
set aval_log1_t;
if upcase(_name_) = "GEOCV" then do;
txtn = 5;
txt = "Geometric Mean (CV%)";
end;
else if upcase(_name_) = "CIGM" then do;
txtn = 6;
txt = "95% CI of Geometric Mean";
end;
run;

data aval_f;
set aval_t n_freq_t aval_log1_t;
run;

proc sort data=aval_f;
by paramn param apuper apuperc avisitn avisit txtn txt;
run;

proc sort data=adqspa;
by trtpn paramn param apuper apuperc avisitn avisit ;
run;

data n_chg;
length txt $200.;
set adqspa;
where pchg = .;
txtn = 2;
txt = "<Missing, n(%)>";
output;
run;

proc sort data=n_chg out=n1_chg nodupkey dupout=dup;
by usubjid paramn avisitn txtn ;
run;

proc freq data=n1 noprint;
tables paramn*param*apuper*apuperc*avisitn*avisit*txtn*txt*trtpn /out=n_chg_freq;
run;

data n_chg_freq;
length countx $50.;
set n_chg_freq;
countx = strip(put(count, best.));
run;

proc sort data=n_chg_freq;
by paramn param apuper apuperc avisitn avisit txtn txt;

```

```

run;

proc transpose data=n_chg_freq out=n_chg_freq_t prefix=chg_;
by paramn param apuper apuperc avisitn avisit txtn txt;
var countx;
id trtpn;
run;

data n_chg_freq_t;
set n_chg_freq_t;
where paramn ne .;
run;

proc means data=adqspa(where=(pchg ne . and ablfl ne "Y")) noprint;
var pchg;
by trtpn paramn param apuper apuperc avisitn avisit ;
output out=chg n =n mean = mean std = std median = median min = min max = max q1 = q1 q3 = q3 lclm = lclm uclm = uclm;
run;

data chg1;
set chg;
length median1 Q2575 Minmax Meansd CIAM n1 $50.;
if not missing(median) then median1 = left(compress(put(round(median,0.1),8.1)));
if not missing(mean) and not missing(std) then meansd = left(compress(put(round(mean,0.1),8.1)) || ' (' || left(compress(put(0.01*ceil(std/0.01),8.2))) || ')');
if not missing(min) and not missing(max) then minmax = left(compress(put(min,8.)) || ', ' || left(compress(put(max,8.))));
if not missing(lclm) and not missing(uclm) then ciam = strip(put(0.1*floor(lclm/0.1),8.1)) || ', ' || strip(put(0.1*ceil(uclm/0.1),8.1));
if not missing(q1) and not missing(q3) then q2575 = strip(put(round(q1,0.1),8.1)) || ', ' || strip(put(round(q3,0.1),8.1)) ;

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

if trtpn= 4 then BigN= &NTHS;
if trtpn= 5 then BigN = &NMCC;

if ^missing(BigN) and ^missing(n) then msng=BigN-n;
if .<msng^=0 then msng_prcnt=(msng/BigN)*100;
if .<msng^=0 then msg_pct=compress(put(msng, best.))||' ('||compress(put(msng_prcnt, 5.1))||')';

run;

proc sort data=chg1;
by paramn param apuper apuperc avisitn avisit ;
run;

proc transpose data=chg1 out=chg_t prefix= chg_;
by paramn param apuper apuperc avisitn avisit ;
var n1 median1 q2575 minmax meansd ciam msg_pct;
id trtpn;
run;

data chg_t;
length txt $200.;
set chg_t;
if upcase(_name_) = "N1" then do;
txtn = 1;
txt = "n";
end;
if upcase(_name_) = "MSG_PCT" then do;
txtn = 1.5;
txt = "Missing, n (%)";
end;
else if upcase(_name_) = "MEDIAN1" then do;
txtn = 7;
txt = "Median";
end;
else if upcase(_name_) = "Q2575" then do;
txtn = 8;
txt = "Q25, Q75";
end;
else if upcase(_name_) = "MINMAX" then do;

```

```

txtn = 9;
txt = "Min, Max";
end;
else if upcase(_name_) = "MEANSD" then do;
txtn = 10;
txt = "Mean (SD)";
end;
else if upcase(_name_) = "CIAM" then do;
txtn = 11;
txt = "95% CI of Mean";
end;
run;

data chg_f;
set chg_t n_chg_freq_t ;
run;

proc sort data=chg_f;
by paramn param apuper apuperc avisitn avisit txtn txt;
run;

proc sort data=aval_f;
by paramn param apuper apuperc avisitn avisit txtn txt;
run;

data final_a;
merge aval_f chg_f(drop=_name_);
by paramn param apuper apuperc avisitn avisit txtn txt;
THS = &NTHS;
mcc = &Nmcc;

if txtn in (2,3,4) then do;

if trt_4 ne " " then num_4 = input(trt_4, best.);
if trt_5 ne " " then num_5 = input(trt_5, best.);
if num_4 ne . then pp_4 = strip(put(round((num_4/thS)*100, 0.1), 15.1));
if num_5 ne . then pp_5 = strip(put(round((num_5/mcc)*100, 0.1), 15.1));
trt_4 = strip(trt_4)||" ("||strip(pp_4)||)";
trt_5 = strip(trt_5)||" ("||strip(pp_5)||)";

if chg_4 ne " " then chgnum_4 = input(chg_4, best.);
if chg_5 ne " " then chgnum_5 = input(chg_5, best.);
if chgnum_4 ne . then ppnum_4 = strip(put(round((chgnum_4/thS)*100, 0.1), 15.1));
if chgnum_5 ne . then ppnum_5 = strip(put(round((chgnum_5/mcc)*100, 0.1), 15.1));
chg_4 = strip(chg_4)||" ("||strip(ppnum_4)||)";
chg_5 = strip(chg_5)||" ("||strip(ppnum_5)||)";

end;

if strip(txt) in ('Geometric Mean (CV%)', '95% CI of Geometric Mean') then delete;
if txtn = 1.5 and cmiss(trt_4, trt_5, chg_4, chg_5)=4 then delete;
if strip(txt)='Mean (SD)' then txtn=5.5;
if strip(txt)='95% CI of Mean' then txtn=6.5;
run;

data final;
set final_a;

if upcase(_name_) = 'MSG_PCT' then do;

array mb {4} trt_4 trt_5 chg_4 chg_5;
do i = 1 to 4;
if mb{i} = '' then mb{i} = '0';
end;
end;

if upcase(_name_) = 'MSG_PCT' and avisitn = 10 then do;

array md {2} chg_4 chg_5;
do i = 1 to 2;
if md{i} ne '' then md{i} = '';
end;
end;

```

```
RUN;
```

```
proc sort data=final; by paramn param apuper apuperc avisitn avisit txtn txt; run;
```

```
proc sql;
create table page as
select distinct apuper, apuperc, paramn, avisitn
from final
order by paramn, apuper, avisitn;
quit;
```

```
data page1;
set page;
by paramn apuper avisitn;
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.paramn = b.paramn and a.avisitn = b.avisitn and a.apuper = b.apuper
order by paramn, apuper,avisitn, txtn;
quit;
```

```
data final_page;
set final_page end=last;
by paramn apuper avisitn txtn;
if last then call symputx("page", page);
run;
```

```
data tflds.&tflno.(keep=apuper apuperc txt txtn avisitn avisit paramn param trt_4 trt_5 chg_4 chg_5);
set final_page;
run;
```

```
/* Standard - leave this */
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;
```

```
/* Standard - macro for paging */
%macro outrtf(blankn=130, 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/dev/Tables/&tflno..rtf" style=t106343 startpage=yes headery=1440 foot
ery=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.)));
call symput('period', strip(apuper));
call symput('param', strip(param));
call symput('N4', strip(put(th, best.)));
call symput('N5', strip(put(mcc, 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 = '$' %if &i=1 %then %do; contents=' ' %end; %else %do; contents='' %end;;
column page paramn apuper avisitn avisit txtn txt ("THSm2.2$(N=&N4)&linebot" trt_4 chg_4 ) ("mCC$(N=&N5)&linebot" trt_5 chg_5)
;
define paramn / order order = internal noprint;
define page / order order = internal noprint;
define avisitn / order order = internal noprint;
define apuper / order order = internal noprint;

define txtn / order order = internal noprint;
define avisit / "Timepoint" order order=internal style={just=left cellwidth=0.9cm} style(header)={just=left} ;
define txt / "Statistic" display style={just=left cellwidth=1.9cm} style(header)={just=left} ;
define trt_4 / "Value" display style={just=c cellwidth=1.2cm} style(header)={just=center} ;
define trt_5 / "Value" display style={just=c cellwidth=1.2cm} style(header)={just=center};
define chg_4 / "% Change(*)" display style={just=c cellwidth=1cm} style(header)={just=center};
define chg_5 / "% Change(*)" display style={just=c cellwidth=1cm} style(header)={just=center};

compute after avisitn;
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 " ";
line "Parameter: &param";
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 mC
C / THS 2.2 Menthol arms or the last assessment prior to 10AM on Day 1 in the SA arm.';
line '';
line 'Appendix 15.3.6.16';
line "Study ID:ZRRM-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 ;

%outtrtf(blankn=25, halfblnk=N);

%m_logchk;

```