

```

/*=====
| Covance Study Number      : 000000106343      |
| Program Name              : t_mnws_fas.sas      |
| Purpose                   : Descriptive stats of MNWS - FAS      |
| Input Data                : ADSL, ADQSNDSND      |
|
| Output Data               : T_15_02_04_56_02, T_15_02_04_56_02_F      |
|
| Macros Called             : m_printto, m_logchk      |
| Originally Performed by   : kpothuri            |
| Date                     : 18MAY2015            |
|
|=====
| Modification History
|-----
| Modified by              :
| Modification Date       :
| Modification Description :
|=====*/
options notes source source2 nofullstimer validvarname=upcase missing=' ' NOQUOTELNMAX spool replace;
ods _all_ close;
ods listing;

%m_printto(route=YES);

*=====;
* START OF PROGRAM CODE      ;
*=====;

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

/* 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;

%let tflno=T_15_02_04_56_02;
%let title1 = Table 15.2.4.56.2 Descriptive Statistics of MNWS Total Scores - FAS;

*N - counts;
proc sql;
    select count(distinct usubjid) into: N1THS from adam.adsl(where=(trt01pn = 4 and FASFL="Y"));
    select count(distinct usubjid) into: N1MCC from adam.adsl(where=(trt01pn = 5 and FASFL="Y"));
    select count(distinct usubjid) into: N1SAA from adam.adsl(where=(trt01pn = 3 and FASFL="Y"));
quit;
%put &N1THS &N1MCC &N1SAA;

data ADQSNDSND_1;
    set adam.ADSL;
    where avisitn in (102,103,104,105,106,130,160,190);
    if an101f1 = "Y" and FASFL="Y" and paramcd="MNWSRWDS" and ablf1 ne "Y";
run;

*Baseline;
data ADQSNDSND_2;
    set adam.ADSL;
    if an101f1 = "Y" and paramcd="MNWSRWDS";
    if ablf1 = "Y" and FASFL = "Y" then do;
        avisitn = 10;
        avisit = "Baseline";
    output;
    end;
run;

data ADQSNDSND;
    set ADQSNDSND_1 ADQSNDSND_2;
run;
proc sort data=ADQSNDSND;

```

```

by trtpn param paramn avisitn avisit;
run;

*BLOQ, ALOQ;
data n;
  length txt $200.;
  set ADQSND;
  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 param paramn avisitn txtn;
run;
proc freq data=n1 noprint;
  tables param*paramn*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 param paramn avisitn avisit txtn txt;
run;
proc transpose data=n_freq out=n_freq_t prefix=trt_;
  by param paramn avisitn avisit txtn txt;
  var countx;
  id trtpn;
run;
data n_freq_t;
  set n_freq_t;
  where paramn ne .;
run;

*value stats;
proc sort data=ADQSND;
  by trtpn param paramn avisitn avisit;
run;
proc means data=ADQSND(where=(aval ne .)) noprint;
  var aval;
  by trtpn param paramn 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;

*figure;
data tflds.T_15_02_04_56_02_F;
  length trt $8;
  set aval;
  keep trtpn trt param paramn avisitn avisit mean lclm uclm;

  if trtpn=3 then trt="SA";
  if trtpn=4 then trt="THSm2.2";
  if trtpn=5 then trt="mCC";
run;

data aval1;
  set aval;

  array a[9] mean std median min max Q1 Q3 LCLM UCLM;
  do i=1 to 9;
    if n<4 then a[i]=.;
  end;

  if lclm ne . then lclmx = 0.01*floor(100*lclm);
  if uclm ne . then uclmx = 0.01*ceil(100*uclm);
  length median1 Q2575 Minmax Meansd CIAM n1 $50.;
  if not missing(median) then median1 = left(compress(put(round(median,0.01),15.2)));

```

```

    if not missing(mean) and not missing(std) then meansd = strip(put(round(mean, 0.01), 15.2))||" ("||strip(put(0.001*ceil(std/0.001),8.3))||")");
    if not missing(min) and not missing(max) then minmax = left(compress(put(round(min, 0.1), 15.1)) || ', ' || left(compress(put(round(max, 0.1), 15.1))));
    if not missing(q1) and not missing(q3) then q2575 = strip(put(round(q1, 0.01), 15.2))||", "||strip(put(round(q3, 0.01), 15.2));
    n1 = left(compress(put(n,8.)));

    if not missing(lclm) and not missing(uclm) then ciam = strip(put(lclmx, 15.2))||", "||strip(put(uclmx, 15.2));
    else if lclm = . and uclm ne . then ciam = "NA, "||strip(put(uclmx, 15.2));
    else if lclm ne . and uclm = . then ciam = strip(put(lclmx, 15.2))||", NA";
    else if lclm = . and uclm = . then ciam = "NA, NA";
run;

proc sort data=aval1;
    by param paramn avisitn avisit;
run;

proc transpose data=aval1 out=aval_t prefix= trt_;
    by param paramn avisitn avisit;
    var n1 median1 q2575 minmax meansd ciam;
    id trtpn;
run;

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

data aval_f;
    set aval_t n_freq_t;
run;
proc sort data=aval_f;
    by param paramn avisitn avisit txtn txt;
run;

*% change stats;
proc sort data=ADQSND;
    by trtpn param paramn avisitn avisit;
run;

proc means data=ADQSND(where=(pchg ne . and ablfl ne "Y")) noprint;
    var pchg;
    by trtpn param paramn 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;

    array a[9] mean std median min max Q1 Q3 LCLM UCLM;
    do i=1 to 9;
        if n<4 then a[i]=.;
    end;

```

```

if lclm ne . then lclmx = 0.01*floor(100*lclm);
if uclm ne . then uclmx = 0.01*ceil(100*uclm);
length median1 Q2575 Minmax Meansd CIAM n1 $50.;
if not missing(median) then median1 = left(compress(put(round(median,0.01),15.2)));
if not missing(mean) and not missing(std) then meansd = strip(put(round(mean, 0.01), 15.2))||" ("||strip(put(0.001*ceil(std/0.001),8.3))||")";
    if not missing(min) and not missing(max) then minmax = left(compress(put(round(min, 0.1), 15.1)) || ', ' || left(compress(put(round(max, 0.1), 15.1)))));
if not missing(q1) and not missing(q3) then q2575 = strip(put(round(q1, 0.01), 15.2))||", "||strip(put(round(q3, 0.01), 15.2));
    n1 = left(compress(put(n,8.)));

if not missing(lclm) and not missing(uclm) then ciam = strip(put(lclmx, 15.2))||", "||strip(put(uclmx, 15.2));
else if lclm = . and uclm ne . then ciam = "NA, "||strip(put(uclmx, 15.2));
else if lclm ne . and uclm = . then ciam = strip(put(lclmx, 15.2))||", NA";
else if lclm = . and uclm = . then ciam = "NA, NA";
run;

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

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

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

data chg_f;
set chg_t;
run;

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

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

*combine value and % change stats;
data final;
merge aval_f chg_f(drop=_name_);
by param paramn avisitn avisit txtn txt;

THS = &N1THS;
mcc = &N1mcc;
sa = &N1saa;

```

```

*percentage for ALOQ;
if txtn=4 then do;
  if trt_3 ne " " then num_3 = input(trt_3, best.);
  if trt_4 ne " " then num_4 = input(trt_4, best.);
  if trt_5 ne " " then num_5 = input(trt_5, best.);
  if num_3 ne . then pp_3 = strip(put(round((num_3/sa)*100, 0.1), 15.1));
  if num_4 ne . then pp_4 = strip(put(round((num_4/thb)*100, 0.1), 15.1));
  if num_5 ne . then pp_5 = strip(put(round((num_5/mcc)*100, 0.1), 15.1));
  trt_3 = strip(trt_3)||" ("||strip(pp_3)||")";
  trt_4 = strip(trt_4)||" ("||strip(pp_4)||")";
  trt_5 = strip(trt_5)||" ("||strip(pp_5)||")";

  if chg_3 ne " " then chgnum_3 = input(chg_3, best.);
  if chg_4 ne " " then chgnum_4 = input(chg_4, best.);
  if chg_5 ne " " then chgnum_5 = input(chg_5, best.);
  if chgnum_3 ne . then ppnum_3 = strip(put(round((chgnum_3/sa)* 100, 0.1), 15.1));
  if chgnum_4 ne . then ppnum_4 = strip(put(round((chgnum_4/thb)*100, 0.1), 15.1));
  if chgnum_5 ne . then ppnum_5 = strip(put(round((chgnum_5/mcc)*100, 0.1), 15.1));
  chg_3 = strip(chg_3)||" ("||strip(ppnum_3)||")";
  chg_4 = strip(chg_4)||" ("||strip(ppnum_4)||")";
  chg_5 = strip(chg_5)||" ("||strip(ppnum_5)||")";
end;
run;

*Missing calculation;
data final_1;
  set final;
  output;

if txtn=1 then do;
  trt_3=input(trt_3,best.);
  trt_4=input(trt_4,best.);
  trt_5=input(trt_5,best.);
  if chg_3 ne "" then chg_3=input(chg_3,best.);
  if chg_4 ne "" then chg_4=input(chg_4,best.);
  if chg_5 ne "" then chg_5=input(chg_5,best.);

  if chg_3 >=4 or chg_4 >=4 or chg_5 >=4 then do;
    if (sa-trt_3)>0 or (thb-trt_4)>0 or (mcc-trt_5)>0 or (thb-chg_4)>0 or (mcc-chg_5)>0 then do;
      trt_3_1=sa-trt_3;
      trt_4_1=thb-trt_4;
      trt_5_1=mcc-trt_5;

      if trt_3_1 ne . then trt_3=strip(put(trt_3_1,8.0)) || " (" || strip(put(round((trt_3_1/sa)*100, 0.1),15.1)) || ")";
      if trt_4_1 ne . then trt_4=strip(put(trt_4_1,8.0)) || " (" || strip(put(round((trt_4_1/thb)*100, 0.1),15.1)) || ")";
      if trt_5_1 ne . then trt_5=strip(put(trt_5_1,8.0)) || " (" || strip(put(round((trt_5_1/mcc)*100, 0.1),15.1)) || ")";

      if chg_3_1 ne . then chg_3=strip(put(chg_3_1,8.0)) || " (" || strip(put(round((chg_3_1/sa)*100, 0.1),15.1)) || ")";
      if chg_4_1 ne . then chg_4=strip(put(chg_4_1,8.0)) || " (" || strip(put(round((chg_4_1/thb)*100, 0.1),15.1)) || ")";
      if chg_5_1 ne . then chg_5=strip(put(chg_5_1,8.0)) || " (" || strip(put(round((chg_5_1/mcc)*100, 0.1),15.1)) || ")";

      txtn=2;
      txt = "Missing, n(%)";
      output;
    end;
  end;
end;
run;

*small N - BLOQ;
data bloq_small (keep=paramn avisitn b_3 b_4 b_5);
  set aval_f;
  if txtn=1;
  b_3=input(trt_3,best.);
  b_4=input(trt_4,best.);
  b_5=input(trt_5,best.);
run;

*percentage for BLOQ;
data final_1a;
  merge final_1 bloq_small;
  by paramn avisitn;

  if txtn=3 then do;

```

```

    if trt_3 ne " " then num_3 = input(trt_3, best.);
    if trt_4 ne " " then num_4 = input(trt_4, best.);
    if trt_5 ne " " then num_5 = input(trt_5, best.);
    if num_3 ne . then pp_3 = strip(put(round((num_3/b_3)*100, 0.1), 15.1));
    if num_4 ne . then pp_4 = strip(put(round((num_4/b_4)*100, 0.1), 15.1));
    if num_5 ne . then pp_5 = strip(put(round((num_5/b_5)*100, 0.1), 15.1));
    trt_3 = strip(trt_3)||" ("||strip(pp_3)||)";
    trt_4 = strip(trt_4)||" ("||strip(pp_4)||)";
    trt_5 = strip(trt_5)||" ("||strip(pp_5)||)";
end;
run;

data final_2;
set final_1a;
if avisit ne "Baseline" then do;
    if missing(chg_3) then do;
        if txtn=1 then chg_3="0";
    end;
    if missing(chg_4) then do;
        if txtn=1 then chg_4="0";
    end;
    if missing(chg_5) then do;
        if txtn=1 then chg_5="0";
    end;
end;
if txtn=2 then do;
    if trt_3="0 (0.0)" then trt_3="0";
    if trt_4="0 (0.0)" then trt_4="0";
    if trt_5="0 (0.0)" then trt_5="0";

    if chg_3="0 (0.0)" then chg_3="0";
    if chg_4="0 (0.0)" then chg_4="0";
    if chg_5="0 (0.0)" then chg_5="0";
end;

array a[6] trt_3 trt_4 trt_5 chg_3 chg_4 chg_5;
array b[6] trt_3 trt_4 trt_5 chg_3 chg_4 chg_5;
do i=1 to 6;
    if txtn=1 and 0<a[i]<4 then do;
        b[i]="NC";
    end;
end;
run;

proc sql;
create table page as
select distinct paramn, avisitn
from final_2
order by paramn, avisitn;
quit;

data page1;
set page;
by paramn avisitn;
if _n_ = 0 then page = 0;
page+ 1;
run;

proc sql;
create table final_page as
select distinct a.*, b.page
from final_2 as a
left join page1 as b
on a.paramn = b.paramn and a.avisitn = b.avisitn
order by paramn,avisitn,txtn;
quit;

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

data tflds.&tfldno.(keep=txt txtn avisitn avisit param paramn trt_3 trt_4 trt_5 chg_3 chg_4 chg_5);
set final_page;
run;
%put &page;

```

```

/* 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('param', strip(param));
        call symput('N3', strip(put(sa, best.)));
        call symput('N4', strip(put(ths, 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 = '$' %if &i=1 %then %do; contents=' ' %end; %else %do; contents='' %end;;;
column page param avisitn avisit txtn txt ("THSm2.2$(N=&N4)&linebot" trt_4 chg_4 ) ("mCC$(N=&N5)&linebot" trt_5 chg_5)
("SA$(N=&N3)&linebot" trt_3 chg_3)
;
define param / order order = internal noprint;
define page / order order = internal noprint;
define avisitn / 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_3 / "Value" display style={JUST=c cellwidth=1.2cm} style(header)={just=center} ;
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_3 / "% Change(*)" display style={JUST=c cellwidth=1.2cm} style(header)={just=center};
define chg_4 / "% Change(*)" display style={just=c cellwidth=1.2cm} style(header)={just=center};
define chg_5 / "% Change(*)" display style={just=c cellwidth=1.2cm} style(header)={just=center};

compute after avisitn;
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 " ";
  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 15.3.6.15';
  line "Study ID: ZRHM-REXA-08-US      Program: &TFLprg      Status: &status" &_blankn.*"\~\~" "&sysdate" &_blankn.*"\~\~" "(Page &i o
f &page)";
endcomp;

run;
%end;
ods rtf close;
ods results on;
ods path sashelp.tmplmst (read);

%mend ;

%outrtf(blankn=36, halfblnk=N);

*=====;
* END OF PROGRAM CODE ;
*=====;

%m_logchk;

```