

```

/*=====
=====*

| Covance Study Number   : 000000106343          |
|
| Program Name           : t_fager_comp.sas       |
|
| Purpose                : Descriptive Statistics of Fagerström Test for Nicotine Dependence Results – PP Set
|
|                               Product Use Time Period: Period 4
|
|
|                               |
|
| Input Data              : ADAM.ADSL, ADAM.ADBX   |
|                               |
|
| Output Data             : T_15_02_04_51_03       |
|
| Macros Called           : %m_printto, %trt, %mmeans, %outrtf, %m_logchk2          |
|
| Originally Performed by : Ranju Gautam          |
|
| Date                    : 21MAY2015             |
|
|=====
=====|

| Modification History          |
|
| Modified by                   :                   |
|
| Modification Date             :                   |
|                               |
|
| Modification Description      :                   |
|
|=====
=====*/

```

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

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

```
%macro trt(pfl= );
```

```
    proc sql;
```

```
        %global trt1 trt2 trt3;
```

```
        select count(distinct usubjid) into: trt1 from adam.adsl(where=(trt01pn = 4 and &pfl.));
```

```
        select count(distinct usubjid) into: trt2 from adam.adsl(where=(trt01pn = 5 and &pfl.));
```

```
        select count(distinct usubjid) into: trt3 from adam.adsl(where=(trt01pn = 3 and &pfl.));
```

```
    quit;
```

```
%mend;
```

```
%let sup_1 = %nrstr({\super (1)});
```

```
data ADQSND;
```

```
    set adam.ADQSND(where=(paramcd in ('FTNDSC')));
```

```
    if abfl='Y' then do; avisit='Baseline'; avisitn=10; end;
```

```
    if avisitn in (10,190) and paramcd in ('FTNDSC');
```

```
run;
```

```
%macro mmeans(pfl=, prd=, class=, var=, out=);
```

```
*get N values for column headers for each period;
```

```
%trt(pfl=&pfl.);
```

```
data ADQSND1;
```

```

set ADQSND;

where &prd.;

if          trtan=4 then trt=1;

else if trtan=5 then trt=2;

else if trtan=3 then trt=3;

run;

proc means data=ADQSND1 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.)));

if trt=1 then do;

```

```

                                if &trt1.=n1 then miss="";
                                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="";
                                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="";
                                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)='MEANSD' then do; statord=3; stat='Mean (SD)'; 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 stat ne "";

run;

%mend;

%mmeans(pfl=(COMPP4FL='Y'), prd=(COMPP4FL='Y' and avisitn in (10 )), class=PARAMCD PARAM avisitn
avisit , var=aval, out=out_p1);

%mmeans(pfl=(COMPP4FL='Y'), prd=(COMPP4FL='Y' and avisitn in (190)), class=PARAMCD PARAM
avisitn avisit , var=aval, out=out_p4);

%mmeans(pfl=(COMPP4FL='Y'), prd=(COMPP4FL='Y' and avisitn in (190)), class=PARAMCD PARAM
avisitn avisit , var=chg, out=out_c4);

```

```

*macro for mild moderate severe per mock;

%macro mmeans(prd=, class=, var=, by=, out=);

*Bring in data from ADBX for Blood COHb - PP Set for each period per Mock;

data ADQSND1;

    set ADQSND;

    where &prd.;

    if          trtan=4 then trt=1;

    else if trtan=5 then trt=2;

    else if trtan=3 then trt=3;

run;


proc means data=adqsnd1 noprint nway;

    &var;

    class &class. trt;

    output out=results02 n=n1;

run;


proc transpose data=results02 out=&out. prefix=trt name=varname;

    by &class.;

    var n1;

    id trt;

run;

```

data &out.;

set &out.;

length stat \$200 r1 r2 r3 \$ 20;

if avalcat1='Mild' then statord1=0.5;

if avalcat1='Moderate' then statord1=1.5;

if avalcat1='Severe' then statord1=2.5;

if avisitn = 10 then shift1="";

if scan(shift1,-1)='Mild' then statord1=1;

if scan(shift1,-1)='Moderate' then statord1=2;

if scan(shift1,-1)='Severe' then statord1=3;

if shift1='Mild to Mild' then do; statord=7; stat='Mild (0 - 3), n(%)'; end;

if shift1='Moderate to Mild' then do; statord=8; stat='Moderate (4 - 6), n(%)'; end;

if shift1='Severe to Mild' then do; statord=9; stat='Severe (7 - 10), n(%)'; end;

if shift1='Mild to Moderate' then do; statord=7; stat='Mild (0 - 3), n(%)'; end;

if shift1='Moderate to Moderate' then do; statord=8; stat='Moderate (4 - 6), n(%)'; end;

if shift1='Severe to Moderate' then do; statord=9; stat='Severe (7- 10), n(%)'; end;

if shift1='Mild to Severe' then do; statord=7; stat='Mild (0 - 3), n(%)'; end;

if shift1='Moderate to Severe' then do; statord=8; stat='Moderate (4 - 6), n(%)'; end;

if shift1='Severe to Severe' then do; statord=9; stat='Severe (7 - 10), n(%)'; end;

if trt1 ne . then r1=strip(put(trt1, best.))||' ('||put(trt1\*100/&TRT1., 4.1)||')';

if trt2 ne . then r2=strip(put(trt2, best.))||' ('||put(trt2\*100/&TRT2., 4.1)||')';

```

if trt3 ne . then r3=strip(put(trt3, best.)) || ' (' || put(trt3*100/ &TRT3., 4.1) || ')';

if statord1 not in (0.5, 1.5, 2.5) then avalcat1 =";

run;

proc sort data= &out.;
    by &by;

run;

%mend;

%mmeans(prd=(COMPP4FL='Y' and avisitn in (10)), class=paramcd param avisitn avisit avalcat1 , var=,
by= avalcat1 statord statord1 , out=out_gr1);

%mmeans(prd=(COMPP4FL='Y' and avisitn in (190)), class=paramcd param avisitn avisit avalcat1, var=,
by= avalcat1 statord , out=out_gr4);

%mmeans(prd=(COMPP4FL='Y' and avisitn in (190)), class=paramcd param avisitn avisit shift1, var=, by=
statord1 statord, out=out_chg_gr4);

*combining the upper part : baseline statitic with the severity part;

data part1;

    set out_p1(in=a) out_gr1(in=b);

    if a then ord=1;

    if b then ord=2;

    if avalcat1 ne " then stat=avalcat1;

    if statord=11 then delete;

```



```
        if statord=2 and r1="" and r2="" and r3="" then delete;
run;
```

```
*combining the lower part;
```

```
*combining the day 90 value and change side by side for statistic;
```

```
proc sort data=out_p4 out=ds1;

        by paramcd param avisitn avisit statord stat varname;

run;
```

```
proc sort data=out_c4 out=ds2;

        by paramcd param avisitn avisit statord stat varname;

run;
```

```
data part2a;

        merge ds1(in=a) ds2(in=b rename=(r1=c1 r2=c2 r3=c3));

        by paramcd param avisitn avisit statord stat varname;

run;
```

```
*combining the day 90 value and change side by side for grade;
```

```
proc sort data=out_gr4 out=gr4;

        by paramcd param avisitn avisit statord1 statord;

run;
```

```

proc sort data=out_chg_gr4 out=chg_gr4;

    by paramcd param avisitn avisit statord1 statord ;

run;

data part2b;

    merge gr4(in=a drop=shift1 varname stat trt1 trt2 trt3 _label_) chg_gr4(in=b drop=avalcat1
rename=(r1=c1 r2=c2 r3=c3));

    by paramcd param avisitn avisit statord1 statord;

    if shift1 ne " then stat=shift1;

    if shift1 eq " and avalcat1 ne " then stat=avalcat1;

    keep paramcd param avisitn avisit r1 r2 r3 c1 c2 c3 stat statord1 statord ;

run;

```

```

data part2;

    set part2a(in=a) part2b(in=b);

    if a then ord=1;

    if b then ord=2;

    IF stat='Mild to Mild' then delete;

    IF stat='Moderate to Moderate' then delete;

    IF stat='Severe to Severe' then delete;

```

```

    if stat ne "";
run;

*create final dataset per mock;

data final;

    set part1(in=p1) part2(in=p4) ;

    if p1 or p4 then period="4";

    if statord1=0.5 then stat='Mild (0 - 3), n(%)';

    if statord1=1.5 then stat='Moderate (4 - 6), n(%)';

    if statord1=2.5 then stat='Severe (7 - 10), n(%)';

    if r1 ne "" or r2 ne "" or r3 ne "" then do;

        if stat='Moderate (4 - 6), n(%)' then do;

            if r3="" then r3='0 ( 0.0)';

        end;

        if stat='Severe (7 - 10), n(%)' then do;

            if r3="" then r3='0 ( 0.0)';

        end;

    end;

end;

```

```

        if stat='Moderate to Mild' then stat=' Moderate to Mild, n(%)';

        if stat='Severe to Mild'   then stat=' Severe to Mild, n(%)';


    if stat='Mild to Moderate' then stat=' Mild to Moderate, n(%)';

        if stat='Severe to Moderate'      then stat=' Severe to Moderate, n(%)';


    if stat='Mild to Severe' then stat=' Mild to Severe, n(%)';

        if stat='Moderate to Severe'      then stat=' Moderate to Severe, n(%)';


run;


*****
,

*create new page for each timepoint for report ;

*****
,

proc sql;

    create table page as

    select distinct paramcd, period, avisitn

    from final

    order by paramcd desc, period, avisitn;

quit;


data page1;

    set page;

    by descending paramcd period 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.paramcd=b.paramcd and a.avisitn=b.avisitn and a.period=b.period

order by page,ord,statord1,statord ;

quit;
```

```
data final_page;

set final_page;

if avisitn=190 then do;

if strip(stat) in ('Moderate (4 - 6), n(%)' 'Mild to Moderate, n(%)' 'Severe to Moderate, n(%)'

'Severe (7 - 10), n(%)', 'Moderate to Severe, n(%)' 'Mild to Severe, n(%)') then page=3;

end;

run;
```

```
proc sort data=final_page;

by paramcd avisitn period page ord statord1 statord;
```

```
run;
```

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

```
    set final_page end=last;
```

```
    by paramcd avisitn period page ord statord1 statord;
```

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

```
run;
```

```
%let tflno=T_15_02_04_51_03;
```

```
data tflds.&tflno(keep=paramcd param avisitn avisit stat THSm1 mCC1 SA1 THSm_chg1 mCC_chg1  
SA_chg1 statord period page);
```

```
    set final_page;
```

```
        length THSm1 mCC1 SA1 THSm_chg1 mCC_chg1 SA_chg1 $ 30;
```

```
    if THSm ne " " then THSm1=THSm;
```

```
    if mCC ne " " then mCC1=mCC;
```

```
    if SA ne " " then SA1=SA;
```

```
    if THSm_chg ne " " then THSm_chg1=THSm_chg;
```

```
    if mCC_chg ne " " then mCC_chg1=mCC_chg;
```

```
    if SA_chg ne " " then SA_chg1=SA_chg;
```

```
run;
```

```
*****.
```

```
*create output report ;
```

```
*****.
```

```
*get N for each period for column header;
```

```
proc sql;
```

```
select count(distinct usubjid) into: N1THS from adam.adsl(where=(trt01pn = 4 and COMPP1FL = "Y"));
```

```
select count(distinct usubjid) into: N1MCC from adam.adsl(where=(trt01pn = 5 and COMPP1FL = "Y"));
```

```
select count(distinct usubjid) into: N1SAA from adam.adsl(where=(trt01pn = 3 and COMPP1FL = "Y"));
```

```
select count(distinct usubjid) into: N2THS from adam.adsl(where=(trt01pn = 4 and pprot2fl = "Y"));
```

```
select count(distinct usubjid) into: N2MCC from adam.adsl(where=(trt01pn = 5 and pprot2fl = "Y"));
```

```
select count(distinct usubjid) into: N2SAA from adam.adsl(where=(trt01pn = 3 and pprot2fl = "Y"));
```

```
select count(distinct usubjid) into: N3THS from adam.adsl(where=(trt01pn = 4 and pprot3fl = "Y"));
```

```
select count(distinct usubjid) into: N3MCC from adam.adsl(where=(trt01pn = 5 and pprot3fl = "Y"));
```

```
select count(distinct usubjid) into: N3SAA from adam.adsl(where=(trt01pn = 3 and pprot3fl = "Y"));
```

```
select count(distinct usubjid) into: N4THS from adam.adsl(where=(trt01pn = 4 and COMPP4FL = "Y"));
```

```
select count(distinct usubjid) into: N4MCC from adam.adsl(where=(trt01pn = 5 and COMPP4FL = "Y"));
```

```
select count(distinct usubjid) into: N4SAA from adam.adsl(where=(trt01pn = 3 and COMPP4FL = "Y"));
```

```
quit;
```

```
options number nodate orientation=landscape missing=' ';
```

```
ods escapechar='$';
```

```
%let linetop = \brdrt\brdrs\brdrw30;
```

```
%let linebot = \brdrb\brdrs\brdrw30;
```

```
* Standard - macro for paging ;
```

```
%macro outrtf(blankn=130, halfblnk=N, dsn=);
```

```
%let title1 = %nrstr(Table 15.2.4.51.3 Descriptive Statistics of Fagerstrom Test for Nicotine Dependence);
```

```
%let title2 = %nrstr(Results - Compliant Population);
```

```
%let TFL_Part=%scan(&_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;

        if _n_ =1 then
            do;

                call symput('title3', "Product Use Time Period: Period " || Period );

                if period="4" then do;

                    call

symput('trt1', strip(put(&N4THS., best.)));

                    call

symput('trt2', strip(put(&N4MCC., best.)));

                    call

symput('trt3', strip(put(&N4SAA., best.)));

                end;

```

```

end;

_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));

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 page avisitn avisit ord stat ("THSm2.2$(N=&trt1)&linebot" THSm
THSm_chg )

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

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

define page / noprint order ;

```

```

define avisitn / group noprint order ;

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


define ord / noprint order ;


define stat /"Statistic" display style={cellwidth=1.9cm asin=on}
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/$Shift &sup_1" display style={JUST=c
cellwidth=1cm} style(header)={just=center};

define mCC_chg /"Change/$Shift &sup_1" display style={just=c
cellwidth=1cm} style(header)={just=center};

define SA_chg /"Change/$Shift &sup_1" display style={just=c cellwidth=1cm}
style(header)={just=center};


break after page/page;


compute after ord;

line " ";

endcomp;


compute before _page_ / style={just=left protectspecialchars=off};

line "\b\fs24\sa24&_FSRTITL." ;

```

```
line "\b\fs24\sa24&title2." ;
```

```
line " ";
```

```
line "Parameter: &param";
```

```
line "\b\fs24\sa24&title3." ;
```

```
line " ";
```

```
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: [1] %change/shift from baseline, where baseline is defined  
as the last assessment prior to first randomized product use in mCC / THS 2.2 Menthol ';
```

```
line 'arms or the last assessment prior to 10 AM on Day 1 in the SA  
arm.';
```

```
line ' ';
```

```
line 'Appendix 15.3.6.13';
```

```
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=36, halfblnk=N);
```

```
ods listing;
```

```
%m_logchk2;
```