

```

/*=====
| Covance Study Number      : 000000106343                                |
| Program Name              : t_proch_comp.sas                            |
| Purpose                   : Descriptive stats of Prochaska 'Stage of Change' Questionnaire - |
|           Compliant              |                                         |
| Input Data                : ADSL, ADQSDND                                |
|
| Output Data               : T_15_02_04_59_02                            |
|
| Macros Called             : m_printto, m_logchk                        |
| Originally Performed by  : kpothuri                                     |
| Date                    : 20MAY2015                                    |
|
|=====
| 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_59_02;
%let title1 = Table 15.2.4.59.2 Descriptive Statistics of Prochaska 'Stage of Change' Questionnaire Results - Compliant;

*N - counts;
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 COMPP2FL = "Y"));
select count(distinct usubjid) into: N2MCC from adam.adsl(where=(trt01pn = 5 and COMPP2FL = "Y"));
select count(distinct usubjid) into: N2SAA from adam.adsl(where=(trt01pn = 3 and COMPP2FL = "Y"));

select count(distinct usubjid) into: N3THS from adam.adsl(where=(trt01pn = 4 and COMPP3FL = "Y"));
select count(distinct usubjid) into: N3MCC from adam.adsl(where=(trt01pn = 5 and COMPP3FL = "Y"));
select count(distinct usubjid) into: N3SAA from adam.adsl(where=(trt01pn = 3 and COMPP3FL = "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;
%put &N1THS &N1MCC &N1SAA;

data ADQSDND_1;
    set adam.ADQSDND;
    where (COMPP1FL = "Y" and 101<=avisitn <= 105) or (COMPP2FL = "Y" and avisitn = 130) or (COMPP3FL = "Y" and avisitn = 160) or (COMPP4FL = "Y" and avisitn = 190);
    if anl01fl = "Y" and paramcd in ("PROCH01", "PROCH02", "PROCH03");
run;

*Baseline to repeat for each period;

```

```

data ADQSND_2;
set adam.ADQSND;
if anl01fl = "Y" and paramcd in ("PROCH01", "PROCH02", "PROCH03");
if ablfl = "Y" and COMPP1FL = "Y" then do;
avisitn = 10;
avisit = "Baseline";
apuper = 1;
apuperc = "Period 1";
output;
end;
if ablfl = "Y" and COMPP2FL = "Y" then do;
avisitn = 10;
avisit = "Baseline";
apuper = 2;
apuperc = "Period 2";
output;
end;
if ablfl = "Y" and COMPP3FL = "Y" then do;
avisitn = 10;
avisit = "Baseline";
apuper = 3;
apuperc = "Period 3";
output;
end;
if ablfl = "Y" and COMPP4FL = "Y" then do;
avisitn = 10;
avisit = "Baseline";
apuper = 4;
apuperc = "Period 4";
output;
end;
run;

data ADQSND;
set ADQSND_1 ADQSND_2;
run;
proc sort data=ADQSND;
by trtpn param paramn apuper apuperc avisitn avisit;
run;

*paramcds counts;
proc sort data=ADQSND;
by trtpn paramn param apuper apuperc avisitn avisit;
run;
proc means data=ADQSND(where=(aval ne .)) noprint;
var aval;
by trtpn param paramn apuper apuperc avisitn avisit;
output out=aval_n n=n;
run;

*paramcd categories of "PROCH01";
proc sort data=ADQSND;
by trtpn paramn param apuper apuperc avisitn avisit aval avalc;
run;
proc means data=ADQSND(where=(aval ne . and paramcd="PROCH01")) noprint;
var aval;
by trtpn param paramn apuper apuperc avisitn avisit aval avalc;
output out=aval1 n=n;
run;

*dummy records for paramcd categories of "PROCH01";
proc sort data=aval1(where=(not missing(param))) out=aval1_1(keep=trtpn param paramn apuper apuperc avisit avisitn) nodupkey;
by trtpn paramn param apuper apuperc avisitn avisit;
run;
data extra1 (drop=i);
set aval1_1;
format avalc $140.;
by trtpn paramn param apuper apuperc avisitn avisit;

do i=1 to 4;
if i=1 then do; avalc='Yes, i currently smoke'; aval=1; end;
else if i=2 then do; avalc='No, i quit within the last 6 months'; aval=2; end;
else if i=3 then do; avalc='No, i quit more than last 6 months ago'; aval=3; end;
else if i=4 then do; avalc='No, i have never smoked'; aval=4; end;
output;
end;
run;

```

```

proc sort data=aval1;
    by trtpn paramn param apuper apuperc avisitn avisit aval avalc;
run;
proc sort data=extra1;
    by trtpn paramn param apuper apuperc avisitn avisit aval avalc;
run;
data stats1;
    merge aval1 extra1;
    by trtpn paramn param apuper apuperc avisitn avisit aval avalc;
run;

*paramcd categories of "PROCH02";
proc sort data=ADQSND;
    by trtpn paramn param apuper apuperc avisitn avisit aval avalc;
run;
data ADQSND_mod;
    set ADQSND;
    where aval ne . and paramcd="PROCH02";
    if avalc = "01" then avalc = "1";
run;
proc means data=ADQSND_mod noprint;
    var aval;
    by trtpn param paramn apuper apuperc avisitn avisit aval avalc;
output out=aval2 n =n;
run;

*paramcd categories of "PROCH03";
proc sort data=ADQSND;
    by trtpn paramn param apuper apuperc avisitn avisit aval avalc;
run;
proc means data=ADQSND(where=(aval ne . and paramcd="PROCH03")) noprint;
    var aval;
    by trtpn param paramn apuper apuperc avisitn avisit aval avalc;
output out=aval3 n =n;
run;

*dummy records for paramcd categories of "PROCH03";
proc sort data=aval3(where=(not missing(param))) out=aval3_1(keep=trtpn param paramn apuper apuperc avisitn avisitn) nodupkey;
    by trtpn paramn param apuper apuperc avisitn avisit;
run;
data extra3 (drop=i);
    set aval3_1;
    format avalc $140.;
    by trtpn paramn param apuper apuperc avisitn avisit;

    do i=1 to 3;
        if i=1 then do; avalc='Yes, within the next 30 days'; aval=1; end;
        else if i=2 then do; avalc='Yes, within the next 6 months'; aval=2; end;
        else if i=3 then do; avalc='No, not thinking of quitting'; aval=3; end;
        output;
    end;
run;
proc sort data=aval3;
    by trtpn paramn param apuper apuperc avisitn avisit aval avalc;
run;
proc sort data=extra3;
    by trtpn paramn param apuper apuperc avisitn avisit aval avalc;
run;
data stats3;
    merge aval3 extra3;
    by trtpn paramn param apuper apuperc avisitn avisit aval avalc;
run;

*combine counts;
data comb_p (drop=_freq_);
    set aval_n stats1(in=a) aval2(in=b) stats3(in=c);

    if a then _type_=1;
    if b then _type_=2;
    if c then _type_=3;
run;

proc sort data=comb_p;
    by paramn param apuper apuperc avisitn avisit aval avalc _type_;
run;

```

```

proc transpose data=comb_p out=aval_t prefix= trt_;
  by paramn param apuper apuperc avisitn avisit aval avalc _type_;
  var n;
  id trtpn;
run;

data aval_f (drop=_name_ _label_);
length txt $200.;
set aval_t;

if trt_3 = . then trt_3 = 0;
if trt_4 = . then trt_4 = 0;
if trt_5 = . then trt_5 = 0;

param = strip(param) || "?";

if _type_ = 1 then do;
  if aval=3 then avalc=trim(left(tranwrd(avalc,' last ','')));
  if find(avalc,'No') then substr(avalc,5,1)='I';
  if find(avalc,'Yes') then substr(avalc,6,1)='I';
  txt = "n (%)" || avalc;
end;
if _type_ = 3 then do;
  txt = "n (%)" || avalc;
end;
if _type_ = 2 then do;
  if avalc in ("1","01") then txt = "n (%)" || strip(avalc) || " time";
  else txt = "n (%)" || strip(avalc) || " times";
end;
if _type_ = 0 then txt = "n";
run;

*percents;
data pt (drop=aval avalc _type_ txt rename=(trt_3=sa_c trt_4=ths_c trt_5=mcc_c));
set aval_f;

if _type_ = 0 then output;
run;

data final (drop=sa_c ths_c mcc_c trt_3 trt_4 trt_5 trt_3_pt trt_4_pt trt_5_pt avalc);
merge pt aval_f;
by paramn param apuper apuperc avisitn avisit;

if _type_ in (1,2,3) then do;
  trt_3_pt= round((trt_3/sa_c)*100, 0.1);
  trt_4_pt= round((trt_4/ths_c)*100, 0.1);
  trt_5_pt= round((trt_5/mcc_c)*100, 0.1);

if trt_3_pt = 100 then trt_3_ = strip(put(trt_3,8.0)) || " (" || strip(put(trt_3_pt,8.0)) || ")";
else if trt_3 ne 0 then trt_3_ = strip(put(trt_3,8.0)) || " (" || strip(put(trt_3_pt,15.1)) || ")";
else if trt_3 = 0 then trt_3_ = strip(put(trt_3,8.0));

if trt_4_pt = 100 then trt_4_ = strip(put(trt_4,8.0)) || " (" || strip(put(trt_4_pt,8.0)) || ")";
else if trt_4 ne 0 then trt_4_ = strip(put(trt_4,8.0)) || " (" || strip(put(trt_4_pt,15.1)) || ")";
else if trt_4 = 0 then trt_4_ = strip(put(trt_4,8.0));

if trt_5_pt = 100 then trt_5_ = strip(put(trt_5,8.0)) || " (" || strip(put(trt_5_pt,8.0)) || ")";
else if trt_5 ne 0 then trt_5_ = strip(put(trt_5,8.0)) || " (" || strip(put(trt_5_pt,15.1)) || ")";
else if trt_5 = 0 then trt_5_ = strip(put(trt_5,8.0));
end;

if trt_3="" and trt_4="" and trt_5="" then do;
  trt_3_=strip(put(trt_3,best.));
  trt_4_=strip(put(trt_4,best.));
  trt_5_=strip(put(trt_5,best.));
end;

if apuper = 1 then do;
  period = "Period 1";
  THS = &N1THS;
  mcc = &N1mcc;
  sa = &N1saa;
end;
else if apuper = 2 then do;
  period = "Period 2";
  ths = &n2ths;
  mcc = &n2mcc;
  sa = &n2saa;

```

```

end;
else if apuper = 3 then do;
period = "Period 3";
ths = &n3ths;
mcc = &n3mcc;
sa = &n3saa;
end;
else if apuper = 4 then do;
period = "Period 4";
ths = &n4ths;
mcc = &n4mcc;
sa = &n4saa;
end;
run;

proc sql;
create table page as
select distinct paramn, apuper, 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.apuper = b.apuper and a.avisitn = b.avisitn
order by paramn, apuper, avisitn, _type_, aval;
quit;

data final_page;
set final_page end=last;
by paramn apuper avisitn _type_ aval;
if last then call symputx("page", page);
run;
proc sort data=final_page; by paramn param apuper apuperc avisitn avisit; run;

data tflds.&tflno.;
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('period', strip(apuperc));
        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 = '$' %if &i=1 %then %do; contents=' ' %end; %else %do; contents='' %end;;;
column page paramn apuper avisitn txt ("THSm2.2$(N=&N4)&linebot" trt_4_) ("mCC$(N=&N5)&linebot" trt_5_)
("SA$(N=&N3)&linebot" trt_3_)
;
define page          / order order = internal noprint;
define paramn        / order order = internal noprint;
define avisitn       / order order = internal noprint;
define apuper        / 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};

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 "Population";
    line " ";
    line "Parameter: &param";
    LINE "Product Use Time Period: &period";
    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 15.3.6.19';
    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, halfb1nk=N);

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

%m_logchk;
```