

```

%_mprintto;
options notes nosource;
proc datasets lib=work nolist memtype=data kill; quit;
%put NOTE:
=====;
%put NOTE: Covance Study Number : 000000106326;
%put NOTE: Client Protocol ID   : ZRHM-PK-05-JP;
%put NOTE: Program Name        : /*t_qsu.sas*/t_cohb.sas;/* 1)  AOB
11Aug2014 */
%put NOTE: Purpose              : table of blood COHb %;
%put NOTE: ;
%put NOTE: Input Data           : ADAM.ADBX;
%put NOTE: Output               : t_15_2_4_8_1(cohb);
%put NOTE: Macros Called        : _MPRINTTO;
%put NOTE: ;
%put NOTE: Programmed by        : cvn_aobyne;
%put NOTE: Creation Date        : 2014-08-08;
%put NOTE: SAS Version          : 9.3;
%put NOTE: ;
%put NOTE: == Latest Run
=====;
%put NOTE: Run by                : &sysuserid;
%put NOTE: Date/Time             :
%sysfunc(putn(%sysfunc(date()),e8601da.))T%sysfunc(putn(%sysfunc(time()),
e86011z.));
%put NOTE: ;
%put NOTE: == Modification History
=====;
%put NOTE: Date      Initials   No. Reason;
%put NOTE: 11Aug2014  AOB        1)  Program header amended;
%put NOTE: 11Aug2014  AOB        2)  Footnotees amended;
%put NOTE: 24Sep2014  KB         3)  Removed options replace;
%put NOTE:
=====;
options notes source source2 nofullstimer validvarname=upcase missing='
';
ods _all_ close;
ods listing;

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

%let tflno=T_15_02_04_08_01(cohb);

%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",""));
run;

*****;

```

```

* read in data ;
*****;

/*Use ADSL to get N values for column headers*/
data adsl;
    set adam.adsl(where=(pprotfl='Y'));
    if analgrln=1 then do;
        if index(trt01a,'THS 2.2') or index(trt02a,'THS 2.2') then
trtord=4;
        output;
        if index(trt01a,'mCC') or index(trt02a,'mCC') then trtord=5;
        output;
    end;
    else if analgrln=2 then do;
        if index(trt01a,'THS 2.2') or index(trt02a,'THS 2.2') then
trtord=6;
        output;
        if index(trt01a,'NRT') or index(trt02a,'NRT') then trtord=7;
        output;
    end;
    else if missing(analgrln) then delete;
run;

proc sort data=adsl nodupkey out=adsl1;
    by analgrln analgr1 trtord subjid;
run;

proc freq data=adsl1(where=(not missing(trtord))) noprint;
    table analgrln*analgr1*trtord/ out =tot(drop=percent
rename=(count=total));
run;

data tot2;
    set tot;
    call symput('trt' || compress(put(trtord,best.)), compress(total));
run;

/*Bring in appropriate data from ADBX*/
data adbx;
    set adam.adbx(where=(anl02fl='Y' and pprotfl='Y' and
paramcd='CARBXHGB'));
run;

data adbx_orig;
    set adbx;

    statval=aval;
run;

proc sort data=adbx_orig;
    by analgrln analgr1 trtan trta atptn atpt;
run;

```

```

proc means data=adbx_orig noprint;
    var statval;
    by analgrln analgrl trtan trta atptn atpt;
    output out=results02 n=n1 mean=mean1 std=std1 median=median1 min=min1
max=max1 q1=q1 q3=q3;
run;

data results03;
    set results02;
    attrib meansd minmax n median quart length=$20.;

    n = left(compress(put(n1,8.)));
    if not missing(median1) then median =
left(compress(put(median1,8.2)));
    if not missing(mean1) and not missing(std1) then meansd =
left(compress(put(round(mean1,0.01),8.2))) || ' (' ||
left(compress(strip(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(q1) and not missing(q3) then quart =
strip(strip(put(0.01*floor(q1/0.01),8.2)) || ', ' ||
strip(put(0.01*ceil(q3/0.01),8.2)));

    drop n1 mean1 std1 median1 min1 max1 q1 q3;
run;
%macro outrtf(blankn=, halfblnk=);

%if &halfblnk=N %then %let halfblnk=;
%else %if &halfblnk=Y %then %let halfblnk=~;

/*Obtain subjects with values BLOQ*/
data adbx_blq;
    set adbx;
    where bloqfl='Y';

    statval=aval;
run;

proc freq data=adbx_blq noprint;
    table analgrln*analgrl*trtan*trta*atptn*atpt/ out
=blq(drop=percent);
run;

%let dsid=%sysfunc(open(blq));
%let nsum=%sysfunc(attrn(&dsid.,nobs));
%let rc=%sysfunc(close(&dsid.));

%put "Check " &nsum.;

%if &nsum. lt 1 %then %do;
    proc sort data=adbx nodupkey out=tpts(keep=analgrln analgrl
trtan trta atptn atpt);
        by analgrln analgrl trtan trta atptn atpt;

```

```

run;

data blq1;
    set tpts;
    attrib blq length=$50.;
    blq='0';
run;

%end;

%else %do;
    data blq1;
        attrib blq length=$50.;
        merge blq(in=a) tot;
        by analgrln analgrl trtord;
        if not a then do;
            count=0;
        end;
        percent=count/total*100;

        if count=0 then blq='0';
        else if percent=100 then blq= put(count,3.)||' (100%)';
        else blq=put(count,3.)||' ('||put(percent,3.)||'%)';
    run;
%end;

```

/*Obtain the geometric mean*/

```

data gmean;
    set adbx_orig;
    statvall=statval;
    ln_statvall=log(statvall);
run;

```

```

proc means data=gmean noprint;
    output out=gmean1 mean=mean std=std1 lclm=lci1 uclm=uci1;
    var ln_statvall;
    by analgrln analgrl trtan trta atptn atpt;
run;

```

```

data gmean2;
    set gmean1;
    gmean1=exp(mean);
    gmean=left(compress(put(gmean1,8.2)));
    gcv=compress(strip(put(0.01*ceil(sqrt(exp(std1*std1)-
1)*100/0.01),8.2)));
    glci=exp(lci1);
    guci=exp(uci1);
    keep analgrln analgrl trtan trta atptn atpt gmean gcv glci guci std1;
run;

```

/*Combine the Gmean and BLQ with other stats*/

```

data results04;

```

```

merge results03 gmean2 blq1;
    attrib gmeancv length=$20.;
by analgrln analgrl trtan trta atptn atpt;
if analgrln=2 and trtan=4 then trtan=6;
    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)));

run;

proc sort data=results04;
    by atptn atpt;
run;

proc transpose data=results04 out=results05 prefix=_ name=varname;
    by atptn atpt;
    var n meansd median minmax ci quart gmeancv blq;
    id trtan;
    idlabel trta;
run;

data results06;
    set results05;
    attrib stat length = $100.;

    if varname='N' then do;
        statord=1;
        stat='n';
    end;
        else if varname='BLQ' then do;
            statord=2;
            stat='BLOQ - n (%)';
        end;
        else if varname='GMEANCV' then do;
            statord=3;
            stat='Geometric Mean (CV%)';
        end;
    else if varname='CI' then do;
        statord=4;
        stat='95% CI';
    end;
    else if varname='MEDIAN' then do;
        statord=5;
        stat='Median';
    end;
    else if varname='QUART' then do;
        statord=6;
        stat='Q25, Q75';
    end;
    else if varname='MINMAX' then do;
        statord=7;

```

```

        stat='Min, Max';
    end;
    else if varname='MEANSD' then do;
        statord=8;
        stat='Mean (SD)';
    end;
    drop varname;
run;

data results07;
    set results06;

    if stat='N' then do;
        if missing(_4) then _4='0';
        if missing(_5) then _5='0';
        if missing(_6) then _6='0';
        if missing(_7) then _6='0';
    end;
run;

data labels;
set results07;
    attrib _4 label = "THS 2.2 Menthol$(N=&trt4)"
           _5 label = "mCC$(N=&trt5)"
           _6 label = "THS 2.2 Menthol$(N=&trt6)"
           _7 label = "NRT gum$(N=&trt6)"
           atpt label= "Formatted timepoint"
           atpt1 label= "Unformatted timepoint";

           atpt1=atpt;

    if index(atpt,'T0') then atpt=tranwrd(atpt,'T0',"T${sub 0}");

    if index(stat,'BLOQ') and _4=0 and _5=0 and _6=0 and _7=0 then
delete;
run;

/*options replace;*/ /* 3) KB 24Sep2014 */
proc sql noprint;
    create table table.T_15_02_04_08_01 as
    select atpt, atpt1, stat, _4, _5, _6, _7
    from labels
    order by atptn, statord;
quit;
/*options noreplace;*/ /* 3) KB 24Sep2014 */

proc sort data=labels;
    by atptn statord;
run;

data paging;
    set labels;
    by atptn statord;

```

```

        if first.atptn then ln=1; /*Amend to look presentable, and avoid
page overflows*/
        else ln+1;
        if ln=1 then page+1;
        call symput("page",compress(put(page,best.)));
run;

options number nodate orientation=landscape papersize=&p_pgsz 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;

ods path stdlib.tl06326 (read) ;
ods results off;
ods rtf toc_data
file="/cvn/projects/prj/data/000000106326/TFL/&TFL_Part./&tflno..rtf"
style=tl06326 startpage=yes headery=1440 footery=1440 ;
ods noproctitle;
%do i=1 %to &page;

title ;
footnote;
%let wd=0;

data comp;
    set paging end=eof;
    where page=&i;

    /* Amend title as needed */
    _firtitl="Table 15.2.4.8.1 Descriptive Statistics of Blood COHb (%)
Continuous Measurements - PK Population";
    _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;

* 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;

```

```

proc report data = comp missing headline headskip missing nowd split =
'$' contents = "&_FSRTITL";
    column page atptn atpt statord stat ("Group-1 PK &linebot" _4 _5)
("Group-2 PK &linebot" _6 _7);

    define page          / order order = internal noprint;
    define atptn         / order order=internal noprint;
    define atpt          / group style={just=left cellwidth=1.5cm}
style(header)={just=center} "Timepoint";
    define statord       / order order = internal noprint;
    define stat          / display style={just=left cellwidth=1.7cm}
style(header)={just=center} "Statistic";
    define _4            / display style={just=c cellwidth=1.5cm}
style(header)={just=center};
    define _5            / display style={just=c cellwidth=1.5cm}
style(header)={just=center};
    define _6            / display style={just=c cellwidth=1.5cm}
style(header)={just=center};
    define _7            / display style={just=c cellwidth=1.5cm}
style(header)={just=center};

    break after page / page;

    compute after atptn;
        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 "&linebot";
    endcomp;

    compute after _page_ / style={just=left protectspecialchars=off
pretext="&linetop."};
        line 'Note: mCC = menthol conventional cigarettes; NRT gum =
Nicotine Replacement Therapy gum; THS = Tobacco Heating System.';
/*      line "Note: Enrolled Not Randomized refers to all subjects
enrolled but not randomized. Overall Safety refers to enrolled subjects
exposed to THS 2.2 Menthol or NRT gum."; *//* 2) AOB 11Aug2014 */
        LINE "Note: Geometric: mean, CV% and 95% confidence interval
(CI) are reported."/* 2) AOB 11Aug2014 */
        line "Note: T${sub 0} = Time of first product use at single
use day.";
        %if &nsum. ge 1 %then %do;
            line 'Note: LOQ = XX %'; /*Update this value if
required*/
        %end;
        line ' ';

```



```

                line 'Appendix 15.3.3.4';
                line "Path: &TFLpath." &_blankn.*"\~\~" "(Page &i of
&page)";
                line "Program Run: &sysdate   &sysuserid   Program
Status: &status";
                endcomp;
run;
%end;
ods rtf close;
ods results on;
ods path sashelp.tmplmst (read);

%mend ;

%outrtf(blankn=70, halfblnk=N);
ods listing;
proc printto print = "&table./T_15_02_04_08_01.lst" new;
run;

proc contents data = table.T_15_02_04_08_01 varnum;
run;
ods listing close;
proc printto ; run;
*=====;
* END OF PROGRAM CODE                               ;
*=====;
```