

```

%_mprintto;
options notes nosource;
proc datasets lib=work nolist memtype=data kill; quit;
%put NOTE:
=====;
%put NOTE: Covance Study Number : 000000106324;
%put NOTE: Client Protocol ID   : ZRHR-REXC-03-EU;
%put NOTE: Program Name        : t_cyp2.sas;
%put NOTE: Purpose              : table decriptive stats of cyp2a6;
%put NOTE: ;
%put NOTE: Input Data           : ADAM.ADBX ADAM.ADSL;
%put NOTE: Output                : t_15_2_4_51(cyp2);
%put NOTE: Macros Called         : _MPRINTTO;
%put NOTE: ;
%put NOTE: Programmed by        : cvn_jhardman;
%put NOTE: Creation Date        : 2014-07-28;
%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: 04Aug2014   JMH       1) Amended footnote;
%put NOTE: 19Aug2014   AOB       2) Amended footnote;
%put NOTE: 15Sep2014   JR        3) Reverted to prevoius baseline
footnote;
%put NOTE: 16Sep2014   JMH       4) Amended paging and proc report to
fit all data on one page;
%put NOTE: 18Sep2014   JR        5) Updated baseline footnote;
%put NOTE: 18Sep2014   KB        6) Amended dual programming table
creation;
%put NOTE: 25Sep2014   JMH       7) Amended BLQ;
%put NOTE: ;
%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_51(cyp2);

%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=(fasfl='Y'));
run;

proc sort data=adsl nodupkey out=adsl1;
    by trt01an trt01a subjid;
run;

proc freq data=adsl1(where=(not missing(trt01an))) noprint;
    table trt01an*trt01a/ out =tot(drop=percent rename=(count=total));
run;

data tot2;
    set tot;
    call symput('trt' || compress(put(trt01an,best.)),
compress(total));
run;

/*Bring in appropriate data from ADBX*/
data adbx1;
    set adam.adbx(where=(anl02fl='Y' and fasfl ='Y' and paramcd in
('CYP2A6')));
run;

data adbx;
    set adbx1;
    if ablf1='Y' then do; avisit='Baseline'; avisitn=100; end;
    if avisit ne 'Baseline' and avisitn lt 101 then delete;
run;

data adbx_orig;
    set adbx;
    statval=aval;
    type='abs';
    output;
    statval=pchg;
    type='pchg';
    output;
run;

proc sort data=adbx_orig;
    by type trtan trta avisitn avisit atptn atpt;
run;

```

```
proc means data=adbx_orig noprint;
  var statval;
  by type trtan trta avisitn avisit atptn atpt;
  output out=results02 n=n1 mean=mean1 std=std1 median=median1 min=min1
max=max1 q1=q1 q3=q3 lclm=lcil uclm=ucil;
run;
```

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

```
  n = left(compress(put(n1,8.)));
  if not missing(median1) then median =
left(compress(put(round(median1,0.001),10.3)));
  if not missing(mean1) and not missing(std1) then meansd =
left(compress(put(round(mean1,0.001),10.3))) || ' (' ||
left(compress(put(0.0001*ceil(std1/0.0001),10.4))) || ')';
  if not missing(min1) and not missing(max1) then minmax =
left(compress(put(round(min1,0.01),10.2))) || ', ' ||
left(compress(put(round(max1,0.01),10.2)));
  if not missing(lcil) and not missing(ucil) then aci =
strip(put(0.001*floor(lcil/0.001),10.3)) || ', ' ||
strip(put(0.001*ceil(ucil/0.001),10.3));
  if not missing(q1) and not missing(q3) then quart =
strip(put(0.001*floor(q1/0.001),10.3)) || ', ' ||
strip(put(0.001*ceil(q3/0.001),10.3));
```

```
  drop /*n1*/ mean1 std1 median1 min1 max1 q1 q3 ucil lcil ; /* 7) JMH
25Sep2014 */
run;
```

```
/*Obtain subjects with values BLOQ*/
```

```
data adbx_blq;
  set adbx;
  where bloqfl='Y';
  statval=aval;
  type='abs';
  output;
  statsval=pchg;
  type='pch';
  output;
run;
```

```
proc freq data=adbx_blq noprint;
  table type*trtan*trta*avisitn*avisit*atptn*atpt/ out
=blq(drop=percent);
run;
```

```
%macro outrtf(blankn=, halfblnk=);
```

```

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

%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_orig nodupkey out=tpts(keep=type avisitn
avisit atptn atpt trtan trta);
        by trtan trta type avisitn avisit atptn atpt;
    run;

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

%else %do;
    /* 7) start JMH 25Sep2014 */
    /* proc sort data=blq(rename=(trtan=trt01an trta=trt01a)); */
    /* by trt01an trt01a; */
    /* run; */
    /**/
    /* data blq1; */
    /* attrib blq length=$50.; */
    /* merge blq(in=a) tot; */
    /* by trt01an trt01a; */
    /* 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,4.1)||'|'%)'; */
    /* rename trt01an=trtan trt01a=trta; */
    /**/
    /* if type='pch' then blq=''; */
    /* */
    /* run; */
    /**/
    /* proc sort data=blq1; */
    /* by paramn param trtan trta type avisitn avisit atptn
atpt; */
    /* run; */
    PROC SORT DATA=ADBX_ORIG NODUPKEY OUT=TPTS(KEEP=TYPE PARAMN PARAM
AVISITN AVISIT ATPTN ATPT TRTAN TRTA);

```

```

        BY TRTAN TRTA TYPE PARAMN PARAM AVISITN AVISIT ATPTN
ATPT;

    RUN;

    DATA BLQTOTS;
        SET RESULTS03 (RENAME=(N1=TOTAL));
        KEEP TYPE PARAM: TR: AVISIT: ATPT: TOTAL;
    RUN;

    PROC SORT DATA=BLQTOTS; BY TYPE TRTAN TRTA PARAMN PARAM
    AVISITN AVISIT; RUN;
    PROC SORT DATA=TPTS; BY TYPE TRTAN TRTA PARAMN PARAM AVISITN
    AVISIT; RUN;

    DATA TOT_BLQ;
        MERGE TPTS BLQTOTS;
        BY TYPE TRTAN TRTA PARAMN PARAM AVISITN AVISIT;
    RUN;

    PROC SORT DATA=TOT_BLQ;
        BY TYPE PARAMN PARAM TRTAN TRTA AVISITN AVISIT ATPTN
ATPT;

    RUN;

    DATA BLQ1;
        ATTRIB BLQ LENGTH=$50.;
        MERGE BLQ(IN=A) TOT_BLQ;
        BY TYPE PARAMN PARAM TRTAN TRTA AVISITN AVISIT ATPTN
ATPT;

        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.) || '
(' || LEFT(STRIP(PUT(ROUND(PERCENT,0.1),5.1))) || '%' );

        IF TYPE='pch' THEN BLQ='';

    RUN;
    /* 7) end JMH 25Sep2014 */
%end;

/*Obtain the geometric mean*/

data gmean;
    set adbx_orig(where=(type='abs'));
    statvall=statval;
    if aval > 0 then ln_statvall=log(statvall);
run;

proc means data=gmean noprint;
    output out=gmean1 mean=mean std=std1 lclm=lci1 uclm=uci1 nmiss=miss;

```

```

    var ln_statvall;
    by trtan trta type avisitn avisit atptn atpt;
run;

data gmean2;
    set gmean1;
    gmean1=exp(mean);
    gmean=left(compress(put(round(gmean1,0.01),8.2))),);
    gcv=compress(put(0.01*ceil((sqrt(exp(std1*std1)-
1)*100)/0.01),8.2))),);
    glci=exp(lcil);
    guci=exp(ucil);
    keep type trtan trta avisitn avisit atptn atpt gmean gcv glci guci std1
miss;
run;

/*Combine the Gmean and BLQ with other stats*/
proc sort data=results03;
    by trtan trta type avisitn avisit atptn atpt;
run;

data results04;
    merge results03 gmean2 blq1;
    attrib gmeancv length=$20.;
    by trtan trta type avisitn avisit atptn atpt;
    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 trtan trta type avisitn avisit atptn atpt;
run;

proc sort data=blq1;
    by trtan trta type avisitn avisit atptn atpt;
run;

data results05;
    merge results04 blq1;
    by trtan trta type avisitn avisit atptn atpt;
run;

proc sort data=results05;
    by type avisitn avisit atptn atpt;
run;

```

```

proc transpose data=results05(where=(type='abs')) out=results06 prefix=r
name=varname;
  by avisitn avisit atptn atpt;
  var n meansd median minmax aci quart blq gmeancv ci;
  id trtan;
  idlabel trta;
run;

proc transpose data=results05(where=(type='pch' and avisitn>100))
out=results06c prefix=c name=varname;
  by avisitn avisit atptn atpt;
  var n meansd median minmax aci quart blq;
  id trtan;
  idlabel trta;
run;

proc sort data=results06;
  by avisitn avisit atptn atpt varname;
run;

proc sort data=results06c;
  by avisitn avisit atptn atpt varname;
run;

data results07;
  merge results06 results06c;
  by avisitn avisit atptn atpt varname;
  attrib stat variable length = $100.;
  varname=upcase(varname);

  variable=compbl(avisit);

  if varname='N' then do;
    statord=1;
    stat='n';
  end;
  else if varname='BLQ' then do;
    statord=2;
    stat='BLOQ - n (%)';
    /*delete; */          * not required for this output; /* 7) JMH
25Sep2014 */
  end;
  else if varname='GMEANCV' then do;
    statord=3;
    stat='Geometric Mean (CV%)';
    delete;
  end;
  else if varname='CI' then do;
    statord=4;
    stat='95% CI';
    delete;
  end;
  else if varname='MEDIAN' then do;

```

```

        statord=7;
        stat='Median';
    end;
    else if varname='QUART' then do;
        statord=8;
        stat='Q25, Q75';
    end;
    else if varname='MINMAX' then do;
        statord=9;
        stat='Min, Max';
    end;
    else if varname='MEANSD' then do;
        statord=5;
        stat='Mean (SD)';
    end;
    else if varname='ACI' then do;
        statord=6;
        stat='95% CI';
    end;
    drop varname;
run;

data results08;
    set results07;

    if stat='n' then do;
        * havent set changes to missing as not expected ;
        if missing(r1) then r1='0';
        if missing(r2) then r2='0';
        if missing(r3) then r3='0';
    end;
run;

data labels;
set results08;
    attrib r1 label = "Raw$value"
           r2 label = "Raw$value"
           r3 label = "Raw$value"
           c1 label = '%Change$(*)'
           c2 label = '%Change$(*)'
           c3 label = '%Change$(*)';

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

        VARIABLE=TRANWRD(VARIABLE,'/',' / '); /* 7) JMH 25Sep2014
*/

    flag=1;
run;

proc sql noprint;
/*    create table table.T_15_02_04_49 as*/
    CREATE TABLE TABLE.T_15_02_04_51 AS /* 6) KB 18Sep2014 */

```



```

        select avisitn, atpt, variable, statord, stat, r1, c1, r2, c2, r3,
c3 /* 7) JMH 25Sep2014 */
        from labels
        order by avisitn, atptn, statord;
quit;

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

data paging;
    set labels;
    by avisitn atptn statord;
    if first.avisitn and ln gt 5/*8*/ then ln=1; /*Amend to look
presentable, and avoid page overflows*/ /* 7) JMH 25Sep2014 */
    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.tl06324 (read) ;
ods results off;
ods rtf toc_data
file="/cvn/projects/prj/data/000000106324/TFL/&TFL_Part./&tflno..rtf"
style=tl06324 startpage=yes headery=1440 footery=1440 ;
ods noproctitle;

%do i=1 %to &page;

title ;
footnote;
%let wd=0;
%let subpage=1/*2*/; /* 4) JMH 16Sep2014 */

%do j=1 %to &subpage;

%let maxpage=%eval(&page*&subpage);

%let npage=%eval(&subpage*&i+&j-&subpage);

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

```

```

/* Amend title as needed */
_firtitl="Table 15.2.4.51 Descriptive Statistics of CYP2A6 Activity
(%) - FAS";
_upcas=(length("Path: &TFLpath.")-
length(compress("Path:&TFLpath.",'ABCDEFGHIJKLMNOPQRSTUVWXYZ')))/2;
len=&blankn.-length("(page &npage of &maxpage)");
if eof then do;
call symput('_FSRTITL', trim(left(_firtitl)));
call symput('_blankn', compress(put(len,best.)));
end;

drop _firtitl _upcas len;
run;

ods proclabel = ' ';
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 =
'$' %if &i=1 and &j=1 %then %do; contents=' ' %end; %else %do;
contents='' %end;;;
column flag page avisitn atptn variable statord stat
/* 4) start JMH 16Sep2014 */
%if &j=1 %then %do; ("THS 2.2$(N=&trt1)&linebot" r1 c1)
("CC$(N=&trt2)&linebot" r2 c2) /*%end;*/
/*%else %if &j=2 %then %do;*/ ("SA$(N=&trt3)&linebot" r3 c3)
%end;;

define flag / order order = internal noprint;
define page / order order = internal noprint;
define avisitn / order order=internal noprint;
define atptn / order order=internal noprint;
define variable / group style={just=left cellwidth=/*2*/1.4cm}
style(header)={just=center} "Timepoint";
define statord / order order = internal noprint;
define stat / display style={just=left
cellwidth=/*3*//*1.5*/1.7cm} style(header)={just=center} "Statistic"; /*
7) JMH 25Sep2014 */
%if &j=1 %then %do;
define r1 / display style={just=c cellwidth=/*2*/2.2cm}
style(header)={just=center};
define c1 / display style={just=c cellwidth=/*2.5*/2.25cm}
style(header)={just=center};
define r2 / display style={just=c cellwidth=/*2*/2.2cm}
style(header)={just=center};
define c2 / display style={just=c cellwidth=/*2.5*/2.25cm}
style(header)={just=center};
/* %end;*/
/* %else %if &j=2 %then %do;*/

```

```

        define r3                / display style={just=c cellwidth=/*2*/2.2cm}
style(header)={just=center};
        define c3                / display style={just=c cellwidth=/*2.5*/2.25cm}
style(header)={just=center};
        %end;
/* 4) end JMH 16Sep2014 */

break before flag / page %if &i=1 and &j=1 %then %do;
contents="&_fsrtitl" %end; %else %do; contents='' %end;;

break after page / page;

compute after variable;
        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: CC = Conventional cigarettes; SA = Smoking
abstinence; THS = Tobacco Heating System.';
        line "Note: * % change from baseline, where baseline is the
last assessment prior to first product use in CC/THS 2.2 arms on Day 1 or
last assessment prior to 06:29 AM in SA arm on Day 1."; /* 5) JR
18Sep2014 */
/*
        line "Note: * % change from baseline, where baseline is
defined as the last assessment prior to 06:29 AM on Day 1.";*/
        /*LINE 'Note: * % change from baseline, where baseline is
defined as the last assessment prior to first product use in CC/THS 2.2
arms or the last assessment prior to 8:00 AM in SA arm.';*/ /* 2) AOB
19Aug2014 */ /* 3) JR 15Sep2014 */
        %if &nsum. ge 1 %then %do;
                line 'Note: LOQ = XX %'; /*Update this value if required*/
        %end;
        LINE "Note: No BLOQ values recorded for this parameter."; /* 7) JMH
25Sep2014 */
        line ' ';
/*
        line 'Appendix 15.3.3.1';*/
                line 'Appendix 15.3.6.15'; /* 1) JMH 04Aug2014 */
        line "Path: &TFLpath." &_blankn.*"\~\~" "(Page &npage of
&maxpage)";
        line "Program Run: &sysdate &sysuserid Program Status:
&status";
endcomp;
run;

```

```

%end;
%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_51.lst" new;
run;

proc contents data = table.T_15_02_04_51 varnum;
run;
ods listing close;
proc printto ; run;
*=====;
* END OF PROGRAM CODE ;
*=====;

```