

/*-----

Covance Study ID : COV-000000106343

Program Name : t_lb_bc_fas.sas

Purpose : Table 15.2.4.32.2.1 (Descriptive Statistics of 8-epi-PGF2± (units) and 11 DTX-B2 (units) â€“ FAS);

Author : cvn_pshe

Date of Creation : 14MAY2015

Input Data : ADAM.ADSL, ADAM.ADBX,

Output Data :

Macros Called :

Modification History

Modified by :

Modification Date :

Modification Description:

-----*/

proc datasets lib=work kill memtype=data nolist;

run;

%m_printto;

options notes nosource;

options replace;

```
options notes source source2 nofullstimer validvarname=upcase missing=' ';
```

```
ods _all_ close;
```

```
ods listing;
```

```
*=====;
```

```
* START OF PROGRAM CODE                                ;
```

```
*=====;
```

```
%let tflno=T_15_02_04_32_02_01;
```

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

```
*****;
```

```
* 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 usubjid trt01an trt01a;

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' || strip(put(trt01an,best.)),strip(put(total,best.)));

run;

/*Bring in parm raw value data from ADBX*/

%macro rawval (parmcd=,parm=, num=);

data adbx;

    set adam.adbx(where=(anl02fl='Y' and fasfl='Y' and anl03fl='Y' and parmcd in ("&parmcd")));

run;

data adbx ;

    set adbx ;

        if abfl = 'Y' then do; avisit='Baseline'; avisitn=98; end;

```

```

        if avisit='SCREENING' and ablfl ='' then delete;

        else if avisit='DAY -2' and ablfl ='' then delete;

        else if avisit='DAY -1' and ablfl ='' then delete;

        else if avisit='DAY 0' and ablfl ='' then delete;

run;

proc sort  data=adbx ;

    by trtan trta avisitn avisit;

run;

proc means data=adbx  noprint;

/*   where ablfl ='Y' or avisitn in (106 130 160 190 191.01);*/

    var aval;

    by trtan trta avisitn avisit;

    output out=bpstat n=n1 mean=mean1 std=sd1 median=median1 min=min1 max=max1 q1=q1 q3=q3
    lclm =lci1 uclm=uci1;

run;

data bpstat1;

    set bpstat;

    attrib meansd minmax n missc median quart length=$20.;

    n = left(compress(put(n1,8.)));

    *for <missing, n(%>;

```

```

                                if trtan=3 then do;

                                                                if &trt3.=n1 then

missc="";

                                                                else

missc=strip(put((&trt3.- n1), 8.)) || ' (' || strip(put(((&trt3.-n1)*100)/&trt3., 8.1)) || ")";

                                                                end;

                                else if trtan=4 then do;

                                                                if &trt4.=n1 then

missc="";

                                                                else

missc=strip(put((&trt4.- n1), 8.)) || ' (' || strip(put(((&trt4.-n1)*100)/&trt4., 8.1)) || ")";

                                                                end;

                                else if trtan=5 then do;

                                                                if &trt5.=n1

then missc="";

                                                                else

missc=strip(put((&trt5.-n1), 8.)) || ' (' || strip(put(((&trt5.-n1)*100)/&trt5., 8.1)) || ")";

                                                                end;

```

```

                                IF NOT MISSING(MEDIAN1) THEN MEDIAN =
LEFT(COMPRESS(PUT(ROUND(MEDIAN1,0.1),10.1)));

```

```

                                IF NOT MISSING(MEAN1) AND NOT MISSING(SD1) THEN meansd =
LEFT(COMPRESS(PUT(ROUND(MEAN1,0.1),10.1))) || " (" || STRIP(PUT(0.01*CEIL(SD1/0.01),10.2)) || ")";

```

```

                                IF NOT MISSING(MIN1) AND NOT MISSING(MAX1) THEN minmax = strip(put(min1, 10.)) || ",
" || strip(put(max1, 10.));

```

```

                                IF NOT MISSING(Q1) AND NOT MISSING(Q3) THEN QUART =
LEFT(COMPRESS(PUT(ROUND(Q1,0.1),10.1))) || ', ' || LEFT(COMPRESS(PUT(ROUND(Q3,0.1),10.1)));;

```

```

                                IF NOT MISSING(LCI1) AND NOT MISSING(UCI1) THEN ACI = STRIP(PUT(0.1*FLOOR(LCI1/0.1),10.1)) || ',
' || STRIP(PUT(0.1*CEIL(UCI1/0.1),10.1));

```

```

drop n1 mean1 sd1 median1 min1 max1 q1 q3 lci1 uci1 _type_ _freq_;

```

```
run;
```

```
proc sort data=bpstat1;
```

```
by trtan trta avisitn avisit;
```

```
run;
```

```
proc transpose data=bpstat1 out=t_bpstat1;
```

```
by trtan trta avisitn avisit;
```

```
var n missc meansd minmax median quart aci;
```

```
run;
```

```
data sa ths mcc;
```

```
length stat rawval $50;
```

```
set t_bpstat1 (drop=trtan rename=( _name_ =stat col1=rawval)) ;
```

```
if trta='SA' then output sa;
```

```
else if trta='THSm2.2' then output ths;
```

```
else if trta='mCC' then output mcc;
```

```
run;
```

```
proc sort data=sa (rename=(rawval=saval)) ;
```

```
by avisitn avisit stat;
```

```
run;
```

```
proc sort data=ths (rename=(rawval=thsvall));
```

```
by avisitn avisit stat;
```

```
run;
```

```
proc sort data=mcc (rename=(rawval=mccval));
```

```
  by avisitn avisit stat;
```

```
run;
```

```
data stat_&parm;
```

```
  merge sa (drop=trta ) ths (drop=trta) mcc;
```

```
    by avisitn avisit stat;
```

```
    if stat='N' then do; stat='n'; sort=1; end;
```

```
    else if stat='MISSC'      then do; stat='Missing, n(%)'; sort=2; end;
```

```
  else if stat='MEANS'D then do; stat='Mean (SD)'; sort=2.2; end;
```

```
    else if stat='ACI' then do; stat='95% CI'; sort=3; end;
```

```
    else if stat='MEDIAN' then do; stat='Median'; sort=4; end;
```

```
    else if stat='QUART' then do; stat='Q25, Q75'; sort=5; end;
```

```
  else if stat='MINMAX' then do; stat='Min, Max'; sort=6; end;
```

```
    if stat='Mean (SD)' then delete;
```

```
  else if stat='95% CI' then delete;
```

```
    order=&num;
```

```
run;
```

```
%mend rawval;
```

```
%rawval (parmcd=UTXB2CRE,parm=dtx, num=1);
```

```

/*Bring in parm raw value data to log scale from ADBX*/

%macro rawval_l (parmcd=,parm=, num=);

data adbx_l;

    set adam.adbx(where=(anl02fl='Y' and fasfl='Y' and anl03fl='Y' and parmcd in ("&parmcd")));

    if nmiss(aval)=0 then aval=log(aval);

run;


data adbx_l ;

    set adbx_l ;

        if ablfl ='Y' then do; avisit='Baseline'; avisitn=98; end;

        if avisit='Screening' and ablfl =" then delete;

        else if avisit='Day -2' and ablfl =" then delete;

        else if avisit='Day -1' and ablfl =" then delete;

run;


proc sort data=adbx_l ;

    by trtan trta avisitn avisit;

run;


proc means data=adbx_l noprint;

    var aval;

    by trtan trta avisitn avisit;

    output out=bpstat_l mean=mean1 std=sd1 lclm =lci1 uclm=uci1 nmiss=miss;

run;

```



```

data bpstat_l;

set bpstat_l;

length gmean gcv $30 glci guci 8;

gmean1=exp(mean1);

if miss=0 then do;

    gmean=left(compress(put(gmean1,10.1)));

    if not missing(sd1) then gcv=strip(put(0.01*ceil((sqrt(exp(sd1*sd1)-1)*100)/0.01),10.2));

    if not missing(lci1) then glci=exp(lci1);

    if not missing(uci1) then guci=exp(uci1);

end;

run;

```

```

data bpstat1_l;

set bpstat_l;

attrib meansd aci length=$20.;

IF MISS=0 THEN DO;

    if not missing(gcv) then meansd=left(trim(gmean)) || ' (' || left(trim(gcv)) || ')';

    else gmeancv=left(trim(gmean));

    if not missing(glci) and not missing(guci) then aci = strip(strip(put(0.1*floor(glci/0.1),10.1)) || ', ' || strip(put(0.1*ceil(guci/0.1),10.1)));

END;

drop mean1 sd1 lci1 uci1 _type_ _freq_;

run;

proc sort data=bpstat1_l;

by trtan trta avisitn avisit;

run;

```

```

proc transpose data=bpstat1_l out=t_bpstat1_l;

    by trtan trta avisitn avisit;

        var meansd aci;

run;

data sa ths mcc;

    length stat rawval $50;

    set t_bpstat1_l (drop=trtan rename=( _name_=stat col1=rawval)) ;

        if trta='SA' then output sa;

        else if trta='THSm2.2' then output ths;

        else if trta='mCC' then output mcc;

run;

proc sort data=sa (rename=(rawval=saval)) ;

    by avisitn avisit stat;

run;

proc sort data=ths (rename=(rawval=thsvall));

    by avisitn avisit stat;

run;

proc sort data=mcc (rename=(rawval=mccval));

    by avisitn avisit stat;

run;

data stat_&parm;

```

```

merge sa (drop=trta ) ths (drop=trta) mcc;

        by avisitn avisit stat;

if stat='MEANS'D then do; stat='Geometric Mean (CV%)'; sort=2.2; end;

        else if stat='ACI' then do; stat='95% CI of Geometric Mean'; sort=3; end;

        order=&num;

run;

%mend rawval_l;

%rawval_l (parmcd=UTXB2CRE,parm=dtx_l, num=1);

data stat_bx ;

    set stat_dtx stat_dtx_l ;

run;

proc sort data=stat_bx;

    by order avisitn avisit sort;

run;

/*Bring in parm percent change data from ADBX*/

%macro pchgval (parmcd=,parm=, num=);

data adbx;

    set adam.adbx(where=(anl02fl='Y' and fasfl='Y' and anl03fl='Y' and paramcd in ("&parmcd")));

    if nmiss(aval)=0 then aval=log(aval);

```

```
run;
```

```
data adbx ;
```

```
set adbx ;
```

```
if ablfl ='Y' then do; avisit='Baseline'; avisitn=98; end;
```

```
if avisit='Screening' and ablfl =" then delete;
```

```
else if avisit='Day -2' and ablfl =" then delete;
```

```
else if avisit='Day -1' and ablfl =" then delete;
```

```
run;
```

```
proc sort data=adbx ;
```

```
by trtan trta avisitn avisit;
```

```
run;
```

```
proc means data=adbx noprint;
```

```
var pchg;
```

```
by trtan trta avisitn avisit;
```

```
output out=pbpstat n=n1 mean=mean1 std=sd1 median=median1 min=min1 max=max1 q1=q1 q3=q3  
lclm =lci1 uclm=uci1;
```

```
run;
```

```
data pbpstat1;
```

```
set pbpstat;
```

```
attrib meansd minmax n missc median quart length=$20.;
```

```
n = left(compress(put(n1,8.)));
```

```

                                if trtan=3 then do;

                                                                if &trt3.=n1 then

missc="";

                                                                else

missc=strip(put((&trt3.- n1), 8.)) || ' (' || strip(put(((&trt3.-n1)*100)/&trt3., 8.1)) || ")";

                                                                end;

                                else if trtan=4 then do;

                                                                if &trt4.=n1 then

missc="";

                                                                else

missc=strip(put((&trt4.- n1), 8.)) || ' (' || strip(put(((&trt4.-n1)*100)/&trt4., 8.1)) || ")";

                                                                end;

                                else if trtan=5 then do;

                                                                if &trt5.=n1

then missc="";

                                                                else

missc=strip(put((&trt5.-n1), 8.)) || ' (' || strip(put(((&trt5.-n1)*100)/&trt5., 8.1)) || ")";

                                                                end;

```

```

                                IF NOT MISSING(MEDIAN1) THEN MEDIAN =
LEFT(COMPRESS(PUT(ROUND(MEDIAN1,0.1),10.1)));

```

```

                                IF NOT MISSING(MEAN1) AND NOT MISSING(SD1) THEN meansd =
LEFT(COMPRESS(PUT(ROUND(MEAN1,0.1),10.1))) || " (" || STRIP(PUT(0.01*CEIL(SD1/0.01),10.2)) || ")";

```

```

                                IF NOT MISSING(MIN1) AND NOT MISSING(MAX1) THEN minmax = strip(put(min1, 10.)) || ",
" || strip(put(max1, 10.));

```

```

                                IF NOT MISSING(Q1) AND NOT MISSING(Q3) THEN QUART =
LEFT(COMPRESS(PUT(ROUND(Q1,0.1),10.1))) || ', ' || LEFT(COMPRESS(PUT(ROUND(Q3,0.1),10.1)));;

```

```

                                IF NOT MISSING(LCI1) AND NOT MISSING(UCI1) THEN ACI = STRIP(PUT(0.1*FLOOR(LCI1/0.1),10.1)) || ',
' || STRIP(PUT(0.1*CEIL(UCI1/0.1),10.1));

```

```

drop n1 mean1 sd1 median1 min1 max1 q1 q3 lci1 uci1 _type_ _freq_;

```

```
run;
```

```
proc sort data=pbpstat1;
```

```
by trtan trta avisitn avisit;
```

```
run;
```

```
proc transpose data=pbpstat1 out=t_pbpstat1;
```

```
by trtan trta avisitn avisit;
```

```
var n missc meansd minmax median quart aci;
```

```
run;
```

```
data psa pths pmcc;
```

```
length stat pchg $50;
```

```
set t_pbpstat1 (drop=trtan rename=( _name_ =stat col1=pchg)) ;
```

```
if trta='SA' then output psa;
```

```
else if trta='THSm2.2' then output pths;
```

```
else if trta='mCC' then output pmcc;
```

```
run;
```

```
proc sort data=psa (rename=(pchg=sapchg));
```

```
by avisitn avisit stat;
```

```
run;
```

```
proc sort data=pths (rename=(pchg=thspchg));
```

```
by avisitn avisit stat;
```

```
run;
```

```

proc sort data=pmcc (rename=(pchg=mccpchg));

  by avisitn avisit stat;

run;

data stat_&parm;

  merge psa (drop=trta) pths (drop=trta) pmcc;

      by avisitn avisit stat;

      if stat='N' then do; stat='n'; sort=1; end;

      else if stat='MISSC'      then do; stat='Missing, n(%)'; sort=2; end;

else if stat='MEANS'D then do; stat='Mean (SD)'; sort=2.2; end;

      else if stat='ACI' then do; stat='95% CI'; sort=3; end;

      else if stat='MEDIAN' then do; stat='Median'; sort=4; end;

      else if stat='QUART' then do; stat='Q25, Q75'; sort=5; end;

      else if stat='MINMAX' then do; stat='Min, Max'; sort=6; end;

      if stat='Mean (SD)' then do; stat='Geometric Mean (CV%)'; sapchg=""; thspchg="";
mccpchg=""; end;

      else if stat='95% CI' then do; stat='95% CI of Geometric Mean'; sapchg=""; thspchg=""; mccpchg=""; end;

      order=&num;

run;

%mend;

%pchgval (parmcd=UTXB2CRE,parm=chgdtx, num=1);

```

```
data stat_bxpchg ;
```

```
    set stat_chgdtx;
```

```
run;
```

```
proc sort data=stat_bxpchg ;
```

```
    by order avisitn avisit sort;
```

```
run;
```

```
data stat;
```

```
    merge stat_bx (drop=stat trta) stat_bxpchg;
```

```
        by order avisitn avisit sort;
```

```
run;
```

```
proc sort data=stat;
```

```
    by order avisitn sort;
```

```
run;
```

```
data stat;
```

```
    set stat;
```

```
        length param $100;
```

```
        if avisit= 'Baseline' and stat ne 'n' then do; sapchg=""; thspchg=""; mccpchg=""; end;
```

```
        if order=1 then param='11-Dehydro-Thromboxane B2 (pg/mg creat)';
```

```
        if avisit='Day 0' then delete;
```

```
        if sort=. then delete;
```



```

        if avisit='Baseline' then do; sapchg=""; mccpchg=""; thspchg=""; end;

run;

* output dataset*;

proc sql noprint;

    create table tflds.&tflno as

    select param as parameter, avisit as timepoint, stat, thsval, thspchg, mccval, mccpchg, saval,
sapchg

    from stat

    order by param, order, avisitn, sort;;

quit;

data paging;

    set stat;

    by order avisitn sort;

    if first.avisitn 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;

data paging;

    set paging;

        by page;

            if first.page then param=param;

            else param="";

```

```
run;
```

```
options number nodate orientation=landscape papersize=Letter /*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;
```

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

```
%if &halfblk=N %then %let halfblk=;
```

```
%else %if &halfblk=Y %then %let halfblk=~;
```

```
ods path stdlib.t106343 (read) ;
```

```
ods results off;
```

```
ods rtf toc_data/* contents*/
```

```
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;
```

```
ods proclabel = ' ';
```

```
data comp;
```

```
set paging end=eof;
```

where page=&i;

/* Amend title as needed */

_firtitl="Table 15.2.4.32.2.1 Descriptive Statistics of 11-DTX-B2 (pg/mg creat) Excluding Assessments within 5 Half-Lives of a Concomitant Medication Affecting the Production of 11-DTX-B2 - FAS";

_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 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 headline headskip nowd split = '#' /*ps = 60 ls = 120*/%if &i=1 %then %do;
contents=' ' %end; %else %do; contents="" %end;;

column order page avisitn param avisit stat

```

("THSm2.2#(N=&trt4)&linebot" thsval thspchg) ("mCC#(N=&trt5)&linebot" mccval
mccpchg) ("SA#(N=&trt3)&linebot" saval sapchg);

```

```

define order / order order = internal noprint;

```

```

define page / order order = internal noprint;

```

```

define avisitn / order order=internal noprint;

```

```

define param / "Parameter (units)" style={just=left cellwidth=2.5cm} style(header)={just=left} ;

```

```

define avisit / group "Timepoint" style={just=left cellwidth=2cm}
style(header)={just=left} ;

```

```

define stat / display "Statistic" style={just=left cellwidth=4.3cm} style(header)={just=left} ;

```

```

define thsval / display "Raw value" style={just=c cellwidth=2.2cm} ;

```

```

define thspchg / display "% Change(*)" style={just=c cellwidth=2.2cm} ;

```

```

define mccval / display "Raw value" style={just=c cellwidth=2.2cm} ;

```

```

define mccpchg / display "% Change(*)" style={just=c cellwidth=2.2cm} ;

```

```

define saval / display "Raw value" style={just=c cellwidth=2.2cm} ;

```

```

define sapchg / display "% Change(*)" style={just=c cellwidth=2.2cm} ;

```

```

break after page / page;

```

```

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 "&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: Percentages are based on the number of subjects indicated in the column header (N).";

        line 'Note: * % change from baseline, where baseline is defined as the last assessment
prior to first randomized product use in mCC / THS 2.2 Menthol arms or the ';

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

                line ' ';

    line "Appendix 15.3.3.1";

        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, halfblk=N);
```

```
ods listing close;
```

```
proc printto ; run;
```

```
%m_logchk;
```

```
*=====;
```

```
* END OF PROGRAM CODE ;
```

```
*=====;
```