

```

%_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_neq.sas;
%put NOTE: Purpose             : table decriptive stats of total NEQ ;
%put NOTE: ;
%put NOTE: Input Data           : ADAM.ADBX ADAM.ADSL;
%put NOTE: Output               : t_15_2_4_34(neq);
%put NOTE: Macros Called        : _MPRINTTO;
%put NOTE: ;
%put NOTE: Programmed by        : cvn_jhardman;
%put NOTE: Creation Date        : 2014-05-15;
%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 BLQ footnote;
%put NOTE: 18Sep2014   JR        2) Amended baseline footnote;
%put NOTE: 25Sep2014   JMH       3) 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_34(neq);

%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
('UNEQ24U' 'UNEQCRE')));
    if ablf1='Y' then do; avisit='Baseline'; avisitn=100; end;
    if avisit ne 'Baseline' and avisitn lt 101 then delete;
run;

data adbx;
    set adbx1;
run;

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

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

proc means data=adbx_orig noprint;
    var statval;
    by type paramn param 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=lci1 uclm=uci1;

```

```

run;

data results03;
  set results02;
  attrib meansd length=$30.
           minmax length=$30.
           n      length=$30.
           median length=$30.
           quart  aci length=$30.;

  n = left(compress(put(n1,8.)));
  if not missing(median1) then median =
left(compress(put(round(median1,0.01),10.2)));
  if not missing(mean1) and not missing(std1) then meansd =
left(compress(put(round(mean1,0.01),10.2))) || ' (' ||
left(compress(put(0.001*ceil(std1/0.001),10.3))) || ')';
  if not missing(min1) and not missing(max1) then minmax =
left(compress(put(round(min1,0.1),10.1))) || ', ' ||
left(compress(put(round(max1,0.1),10.1)));
  if not missing(lcil) and not missing(ucil) then aci =
strip(put(0.01*floor(lcil/0.01),10.2)) || ', ' ||
strip(put(0.01*ceil(ucil/0.01),10.2));
  if not missing(q1) and not missing(q3) then quart =
strip(put(0.01*floor(q1/0.01),10.2)) || ', ' ||
strip(put(0.01*ceil(q3/0.01),10.2));

  drop /*n1*/ mean1 std1 median1 min1 max1 q1 q3 ucil lcil ; /* 3) 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*paramn*param*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 paramn
param avisitn avisit atptn atpt trtan trta);
        by paramn param trtan trta type avisitn avisit atptn
atpt;
    run;

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

%else %do;
    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;

/* 3) start JMH 25Sep2014 */
    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 /*tot(rename=(trt01an=trtan
trt01a=trta))*/BLQTOTS;
        by TYPE trtan trta PARAMN PARAM AVISITN AVISIT;
    run;
/* 3) end JMH 25Sep2014 */

    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;
IF TOTAL NE 0 THEN percent=count/total*100; /* 3) JMH
25Sep2014 */
ELSE PERCENT=0; /* 3) JMH 25Sep2014 */

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;
%end;

/*Obtain the geometric mean*/

data gmean;
    set adbx_orig(where=(type='abs'));
    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 paramn param trtan trta type avisitn avisit atptn atpt;
run;

data gmean2;
    set gmean1;
    gmean1=exp(mean);
    gmean=left(compress(put(round(gmean1,0.01),10.2)));
    gcv=compress(put(0.01*ceil((sqrt(exp(std1*std1)-1)*100)/0.01),10.2));
    glci=exp(lci1);
    guci=exp(uci1);
    keep type paramn param trtan trta avisitn avisit atptn atpt gmean gcv
    glci guci std1;
run;

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

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

data results04;

```

```

merge results03 gmean2 blq1;
  attrib gmeancv length=$30.;
by paramn param 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),10.2)) || ', ' ||
strip(put(0.01*ceil(guci/0.01),10.2)));
run;

proc sort data=results04;
  by paramn trtan trta type avisitn avisit atptn atpt;
run;

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

data results05;
  merge results04 blq1;
  by paramn trtan trta type avisitn avisit atptn atpt;
  if type='abs' then aci='';
run;

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

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

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

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

```

```

data results07;
  merge results06 results06c;
  by paramn avisitn avisit atptn atpt varname;
  attrib stat variable var2 length = $100.;
  varname=upcase(varname);

  if not missing(atpt) then variable=compbl(avisit ||', '|| atpt);
  else variable=compbl(avisit);

  if index(param,'creat') then do;
    var2='Concentration Adjusted for Creatinine ('||scan(param,2,'(')
;
    paramn=paramn+90;
  end;
  else do;
    var2='Quantity Excreted over 24 hours ('||scan(param,2,'(') ;
  end;

  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='Geometric 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;
  else if varname='ACI' then do;
    statord=9;
    stat='95% CI';
  end;
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}');

    flag=1;
run;

proc sql noprint;
    create table table.T_15_02_04_34 as
    select paramn, var2, avisitn, atpt, variable, statord, stat, r1,
c1, r2, c2, r3, c3
    from labels
    order by paramn, var2, avisitn, atptn, statord;
quit;

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

data paging;
    set labels;
    by paramn avisitn atptn statord;
    if (first.avisitn or first.atptn) or ln > 16 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.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;

%let count=0;

%do j=1 %to 2;

%do i=1 %to &page;

%let count=%eval(&count+1);

%let totpage=%eval(&page*2);

title ;
footnote;
%let wd=0;

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

    /* Amend title as needed */
    _firtitl="Table 15.2.4.34 Descriptive Statistics of Urinary Total NEQ
- FAS";
    _upcas=(length("Path: &TFLpath.")-
length(compress("Path:&TFLpath.",'ABCDEFGHIJKLMNOPQRSTUVWXYZ')))/2;
    len=&blankn.-length("(page &count of &totpage)");
    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 =
'$' %if &i=1 %then %do; contents=' ' %end; %else %do; contents='' %end;;
    column flag page paramn var2 avisitn atptn variable statord stat
    %if &j=1 %then %do; ("THS 2.2$(N=&trt1)&linebot" r1 c1)
    ("CC$(N=&trt2)&linebot" r2 c2)%end; %else %do;
    ("SA$(N=&trt3)&linebot" r3 c3)%end;;

    define flag          / order order = internal noprint;
    define page          / order order = internal noprint;
    define paramn        / order order = internal noprint;
    define var2          / group style={just=left cellwidth=2.5cm}
style(header)={just=center} "Variable";
    define avisitn       / order order=internal noprint;
    define atptn         / order order=internal noprint;
    define variable      / group style={just=left cellwidth=1.35cm}
style(header)={just=center} "Timepoint";
    define statord       / order order = internal noprint;
    define stat          / display style={just=left cellwidth=3.7cm}
style(header)={just=center} "Statistic";
    %if &j=1 %then %do;
    define r1            / display style={just=center cellwidth=1.8cm}
style(header)={just=center};
    define r2            / display style={just=center cellwidth=1.8cm}
style(header)={just=center};
    define c1            / display style={just=center cellwidth=1.8cm}
style(header)={just=center};
    define c2            / display style={just=center cellwidth=1.8cm}
style(header)={just=center};
    %end;
    %else %do;
    define r3            / display style={just=center cellwidth=2cm}
style(header)={just=center};
    define c3            / display style={just=center cellwidth=2cm}
style(header)={just=center};
    %end;

break before flag / page %if &i=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: Geometric: mean, CV% and 95% confidence interval
(CI) are reported.';
        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."; /* 2) 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: Collection over 24 hours starting on the Day stated in
the table.";
        %if &nsum. ge 1 %then %do;
            /*line "Note: BLOQ = number of observations imputed using
half limit of quantification (10.0 pg/mL)."; */
            LINE "Note: BLOQ = number of observations imputed using half
limit of quantification for one or more component variables."; /* 1) JMH
04Aug2014 */
        %end;
        line ' ';
        line 'Appendix 15.3.3.2';
        line "Path: &TFLpath." &_blankn.*"\~\~" "(Page &count of
&totpage)";
        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_34.lst" new;
run;

proc contents data = table.T_15_02_04_34 varnum;
run;
ods listing close;
proc printto ; run;
*=====;
* END OF PROGRAM CODE ;
*=====;

```

