

```

%_mprintto;
options notes nosource;
proc datasets lib=work nolist memtype=data kill; quit;
/* 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",""));
run;
%put NOTE:
=====;
%put NOTE: Covance Study Number : 000000106326;
%put NOTE: Client Protocol ID   : ZRHR-PK-05-JP;
%put NOTE: Program Name        : tl_anlpk431.sas;
%put NOTE: Purpose              : table and figure of sensitivity
analysis PK data;
%put NOTE: ;
%put NOTE: Input Data           : ADAM.ADPP;
%put NOTE: Output               : L_15_04_03_01(PK) T_15_2_4_03_01(PK) ;
%put NOTE: Macros Called        : _MPRINTTO;
%put NOTE: ;
%put NOTE: Programmed by        : cvn_ahall;
%put NOTE: Creation Date        : 2014-18-06;
%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: 08Aug2014  AMH         1) CEnter output;
%put NOTE: 08Aug2014  AMH         2) Update listing to match PK02;
%put NOTE:
=====;
options notes source source2 nofullstimer validvarname=upcase missing=' '
NOQUOTELNMAX/*turn off warnings about quoted strings to long*/;;
ods _all_ close;
ods listing;

/*formats macro and appendix output macros*/
%include
"/cvn/projects/prj/development/000000106326/dev/adhoc/TMPLTMIX.sas";

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

```

```

/* Calculate totals for products */

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
colord=1;
    output;
    if index(trt01a,'CC') or index(trt02a,'CC') then colord=2;
    output;
  end;
  else if analgrln=2 then do;
    if index(trt01a,'THS 2.2') or index(trt02a,'THS 2.2') then
colord=1;
    output;
    if index(trt01a,'NRT') or index(trt02a,'NRT') then colord=2;
    output;
  end;
  else if missing(analgrln) then delete;
run;

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

proc freq data=adsl1(where=(not missing(colord))) noprint;
  table analgrln*analgr1*colord/ out =totals2(drop=percent
rename=(count=total));
run;

data _null_;
  set totals2;
  call
symput('tot'||strip(put(colord,best.))||strip(put(analgrln,best.)),strip(
put(total,best.)));
run;

*****;
* read in data ;
*****;
proc sort data=adam.adpp(where=(
paramcd in ('TMAX') /*cmax to count excluded subjects*/
and aval not in (.,0) and crit3fl ne 'Y' AND pprotfl='Y'))
  out=adpp;
  by subjid paramcd;
run;

proc sql;
create table adpp1 as
select *, count(distinct trtan) as trtcount
from adpp
group by analgrln, paramcd, subjid

```

```
order by subjid;  
quit;
```

```
/* take logs and exclude subjects with <2 timepoints*/
```

```
data adpp2 noanal;  
  attrib treat length=$7;  
  set adpp1;  
  if paramcd='TMAX' then param=tranwrd(strip(tranwrd(param,'max','`{sub  
max}')), 'T', 't');  
  else delete;  
  treat=trta;  
  if trtcount>1 then do;  
    logaval=log(aval);  
    output adpp2;  
  end;  
  else output noanal;  
run;
```

```
%fmt(datain=adpp2, start=paramn, label=param, name=param);
```

```
data diff lsmean seed;  
  set dsdtm.tmaxboot;  
  if not missing(seed) then output seed;  
  else if not missing(_treat) then output diff;  
  else output lsmean;  
run;
```

```
proc sort data=diff;  
  by analgrln paramn treat replicate;  
run;
```

```
proc univariate data=diff noprint;  
  by analgrln paramn treat;  
  var difference;  
  output out=diffsbd mean=difference median=med pctlpts=2.5 97.5  
pctlpre=P;  
run;
```

```
proc univariate data=diff noprint;  
  by analgrln paramn treat;  
  var rootmse;  
  output out=diffsbe mean=rootmse;  
run;
```

```
data diffsb;  
  merge diffsbd diffsbe;  
  by analgrln paramn treat;  
run;
```

```
proc sort data=lsmean;  
  by analgrln paramn treat;  
run;
```

```

proc univariate data=lsmean noprint;
  by analgrln paramn treat;
  var lsmean;
  output out=lsmeansb mean=lsmean median=med pctlpts=2.5 97.5 pctlpre=P
;
run;

proc sort data=adpp2;
  by analgrln paramn treat;
run;

proc univariate data=adpp2 noprint;
  by analgrln paramn treat;
  var aval;
  output out=num n=n1;
run;

/*Manipulate datasets for output all relevent stats on each row*/
/*_____*/
data tabout;
  length out $100 stat $100;
  set lsmeansb(in=a) diffsb(in=b) num(in=c);
  /*ordering columns of treatments*/
  if b then colord=3;
  else if treat='THS 2.2' then colord=1;
  else colord=2;

  /* N row*/
  if c then do;
    ord=1;
    stat='n';
    out=compress(put(n1,best.));
    output;
  end;

  if a or b then do;
    /*Back transformation*/
    if a then estimatee=exp(lsmean);
    if b then estimatee=exp(difference);
    lowere=exp(p2_5);
    uppere=exp(P97_5);
    geocv=100*sqrt(exp(rootmse**2)-1);
  /*Gmean (CV%) row*/
    ord=2;
    stat='Geometric LS Mean (CV%)';
    if colord=3 then
out=compress(put(round(100*estimatee,0.01),8.2))||'
('||compress(put(ceil(geocv*100)/100,8.2))||')';
      else out=compress(put(round(estimatee,0.01),8.2));
    output;
  /*95% CI row*/
    ord=3;

```

```

        stat='95% CI';
        if colord=3 then
out=compress(put(floor(100*100*lowere)/100,8.2))||',
'||compress(put(ceil(100*100*uppere)/100,8.2));
        else out=compress(put(floor(100*lowere)/100,8.2))||',
'||compress(put(ceil(100*uppere)/100,8.2));
        output;
/*Median row*/
        ord=4;
        stat='Median';
        if colord=3 then out=compress(put(round(exp(med)*100,0.01),8.2));
        else out=compress(put(round(exp(med),0.01),8.2));
        output;
    end;

run;

/*transpose for output*/
proc sort data=tabout;
    by analgrln paramn ord colord;
run;

proc transpose data=tabout out=ttabout(drop=_NAME_) prefix=col;
    by analgrln paramn ord stat;
    id colord;
    var out;
run;

/* Standard - macro for paging */
%macro outrtf(blankn=67, halfblnk=N, ref=);

/* treatment column headers and footnotes */
/*group 1*/
%let col11=THS 2.2 Menthol#(N=&tot11);
%let col21=mCC#(N=&tot21);
%let col31=THS 2.2 Menthol:mCC#Ratio (%);
%let foot1=%str(mCC = menthol conventional cigarettes);
%let foot11=mCC;

/*group 2*/
%let col12=THS 2.2 Menthol#(N=&tot12);
%let col22=NRT gum#(N=&tot22);
%let col32=THS 2.2 Menthol#:NRT gum Ratio (%);
%let foot2=%str(NRT gum = Nicotine Replacement Therapy gum);
%let foot12=NRT gum;

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

```

```

/* Standard - just change the number to match the listing you're working
on. Also change the letters in the*/
/* bracket, eg ccb = current cigarette brands. Make sure to do this at
the top of the code too. */

    %let tflno=T_15_02_04_03_01(PK);

/* 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",""));
run;

/*page numbers*/
data paging;
    set ttabout;
    by analgrln paramn ;
    flag=1;
    retain ln 0 page 0;
    if first.paramn then ln+1;
    if first.analgrln or ln>3 then do;
        page+1;
        ln=1;
    end;
    if last.analgrln then call symput("tpage",compress(page,best.));
run;

/* Standard - leave this */
options number nodate orientation=landscape papersize=&p_pgsize missing='
' NOQUOTELNMAX/*turn off warnings about quoted strings to long*/;
ods escapechar='`';
%let linetop = \brdrt\brdrs\brdrw30;  * needs to be 1.5pt so calculated
in twips (1/20 pt) ;
%let linebot = \brdrb\brdrs\brdrw30;
%let linebot2 = \brdrb\brdrs\brdrw15;

ods path stdlib.tl06326 (read) ;
ods results off;
ods rtf toc_data/* contents*/
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 &tpage;

ODS PROCLABEL = ' ';
title ;
footnote;

```

```

%let wd=0;

data comp;
  set paging end=eof;
  by paramn ord;
  where page=&i;
  call symput('grp',compress(put(analgrln,best.)));
  /* Amend title as needed */
  _firtitl="Table 15.2.4.3.1 Analysis of Pharmacokinetic
Parameters of Nicotine by Bootstrapping Techniques - PK Population";

  _upcas=(length(_firtitl)-
length(compress(_firtitl,'ABCDEFGHIJKLMNOPQRSTUVWXYZ')))/2;
  len=&blankn.-length("(Page &i of &tpage)");
  if eof then do;
    call symput('_FSRTITL', trim(left(_firtitl)));
    call symput('_blankn', compress(put(len,best.)));
    TITLE1="Table 15.2.4.3.1 Analysis of Pharmacokinetic
Parameters of Nicotine by Bootstrapping Techniques";
    TITLE2="- PK Population";
    CALL SYMPUT('TITLE1', TRIM(LEFT(TITLE1)));
    CALL SYMPUT('TITLE2', TRIM(LEFT(TITLE2)));
  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 missing nowd spanrows split =
'#'
%IF &I=1 %THEN %DO; CONTENTS=' ' %END; %ELSE %DO; CONTENTS='' %END;;
  column flag page paramn ord stat ("Group-&grp PK &linebot." coll
col2 col3) ;
  define flag / order noprint;
  define page / order order = internal noprint;
  define paramn / group style={just=left cellwidth=2cm} "Variable"
format=param.;
  define ord / order order=internal noprint;
  define stat / display style={just=left cellwidth=2.5cm}
"Statistic";
  define coll / display style={just=c/*d*/ cellwidth=2cm}
style(header)={just=center} "&&coll&grp"; /* 2) AMH 08Aug2014 */
  define col2 / display style={just=c/*d*/ cellwidth=2cm}
style(header)={just=center} "&&col2&grp";

```

```

        define col3          / display style={just=c/*d*/ cellwidth=2cm}
style(header)={just=center} "&&col3&grp";

        break after page / page;

break before flag / page %IF &I=1 %THEN %DO;
        CONTENTS="&_FSRTITL" %END; %ELSE %DO; CONTENTS='' %END;;

        compute before page / style={protectspecialchars=off};
        line "&linetop";
        endcomp;

        compute before _page_ / style={just=left protectspecialchars=off};
        line "\b\fs24\sa24 &TITLE1."; /* \b = bold, \fs24 is font size
12pt, \sa24 is space after 12pt */;
        line "\b\fs24\sa24 &TITLE2."; /* \b = bold, \fs24 is font size
12pt, \sa24 is space after 12pt */;
        line "&linebot";
        endcomp;

        compute after paramn ;
        line " ";
        endcomp;

        compute after _page_ / style={just=left protectspecialchars=off
pretext="&linetop."};
        line "Note: &&foot&grp; THS = Tobacco Heating System.";
        line "Note: Point and 95% interval estimates by means of the
percentile bootsrap technique of the adjusted geometric least squares
means from an ANOVA model.";
        line "";
        line "Appendix &ref.";
        line "Path: &TFLpath." &_blankn.*"\~\~" "(Page &i of
&tpage)";
        line "Program Run: &sysdate &sysuserid Program Status:
&status";
        endcomp;

run;
%end;
/*ods rtf close;*/
ods results on;
/*ods path sashelp.tmplmst (read);*/
ods path reset;

%mend ;

%outrtf(blankn=67, halfblnk=N, ref=15.4.4.3.1 and 15.3.3.1);

*****;
*Listing Output;
*****;
/*ordering columns of treatmnts*/

```



```

data listout;
  set diff(in=a) lsmean;
  if a then do;
    colord=3;
    out=compress(put(round(100*exp(difference),0.01),8.2));
    output;
    colord=4;
    out=compress(put(ceil((100*sqrt(exp(rootmse**2)-1))*100)/100,8.2));
    output;
  end;
  else if treat='THS 2.2' then do;
    out=compress(put(round(exp(lsmean),0.01),8.2));
    colord=1;
    output;
  end;
  else do;
    out=compress(put(round(exp(lsmean),0.01),8.2));
    colord=2;
    output;
  end;
run;
/*****
*****/
/*New Listing Output Code 2) AMH 08Aug2014 */
/*****
*****/

/*transpose for output*/
proc sort data=listout;
  by analgr1n paramn replicate;
run;

proc transpose data=listout out=tlistout1(drop=_name_) prefix=col;
  by analgr1n paramn replicate;
  id colord;
  var out;
run;

proc sql;
create table tlistout as
select a.paramn, a.replicate, a.col1, a.col2, a.col3, b.col1 as col4,
b.col2 as col5,
      b.col3 as col6
from tlistout1 as a, tlistout1 as b
where a.analgr1n=1 and b.analgr1n=2 and a.replicate=b.replicate;
quit;

data tmaxin;

```

```

set adpp1;
where trtcoun>1;
num=1;
output;
if trtstmf='S' then aval=aval+0.5;
num=2;
output;
run;
proc sort data=tmaxin; by analgrln trtan num ; run;

/*counts*/
proc freq data=tmaxin(where=(num=1)) noprint;
    table analgrln*trtan/ out =totals1(drop=percent
rename=(count=total));
    table analgrln*trtan*trtstmf/ out
=totals2(where=(trtstmf='S')drop=percent rename=(count=total));
run;

/*correct for zero counts*/

data totals2;
merge totals1(drop=total) totals2;
by analgrln trtan;
if missing(total) then total=0;
run;

proc univariate data=tmaxin noprint;
by analgrln trtan num ;
var aval;
output out=tmaxmed1 median=med;
run;

proc transpose data=tmaxmed1 out=ttmaxin;
by analgrln trtan;
id num;
var med;
run;

data percent;
set ttmaxin;
perc=100*(_1-_2)/_1;
run;

data pagel;
length out $100 stat $100;
set totals1(in=a) totals2(in=b) tmaxmed1(in=c) percent(in=d);
paramn=6;
page=1;
/*ordering columns of treatments*/
if analgrln=1 and trtan=4 then colord=1;
else if analgrln=1 and trtan ne 4 then colord=2;

```

```

else if analgr1n=2 and trtan=4 then colord=3;
else colord=2;
colord=analgr1n+colord-1;
/* first N row*/
if a then do;
    ord=1;
    stat='n';
    out=compress(put(total,best.));
    output;
end;
/* number imputed values*/
if b then do;
    ord=2;
    stat='n (values imputed at T`{sub 0})';
    out=compress(put(total,best.));
    output;
end;
/* number imputed values at 30 sec*/
if c and num=1 then do;
    ord=3;
    stat='Median (imputation by 30 sec)';
    out=compress(put(round(med,0.01),8.2));
    output;
end;
/* number imputed values at 0 sec*/
if c and num=2 then do;
    ord=4;
    stat='Median (imputation by 0 sec)';
    out=compress(put(round(med,0.01),8.2));
    output;
end;
/* number imputed values at 0 sec*/
if d then do;
    ord=5;
    stat='% difference in medians (30 sec - 0 sec)';
    out=compress(put(round(perc,0.01),8.2));
    output;
end;

run;

/*transpose for output*/
proc sort data=page1; by page paramn ord stat; run;

proc transpose data=page1 out=page11(drop=_NAME_) prefix=col;
    by page paramn ord stat;
    id colord;
    var out;
run;

/* Standard - macro for paging */
%macro outrtf(blankn=67, halfblnk=N);

/* treatment column headers and footnotes */

```

```

/*group 1*/
%let col11=THS 2.2 Menthol (N=&tot11);
%let col21=mCC (N=&tot21);
%let col31=THS 2.2 Menthol:mCC Ratio (%);
%let foot1=%str(mCC = menthol conventional cigarettes);
%let foot11=mCC;
/*group 2*/
%let col12=THS 2.2 Menthol (N=&tot12);
%let col22=NRT gum (N=&tot22);
%let col32=THS 2.2 Menthol:NRT gum Ratio (%);
%let foot2=%str(NRT gum = Nicotine Replacement Therapy gum);
%let foot12=NRT gum;

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

/* Standard - just change the number to match the listing you're working
on. Also change the letters in the*/
/* bracket, eg ccb = current cigarette brands. Make sure to do this at
the top of the code too. */

%let tflno=L_15_04_04_03_01(PK);

/* 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",""));
run;

/*page numbers*/
data paging;
    set tlistout;
    by analgrln paramn ;*/
    flag=1;
    retain ln 0 page 1;
    ln+1;
    if ln>20 then do;
        page+1;
        ln=1;
    end;
    call symput("tpage",compress(put(page,best.)));
run;

data _null_;
    set seed;
    call symput('seed',put(seed,8.0));
run;

```

```

data paging;
set /*page11*/ paging;
flag=1;
if page=&tpage and ln>16 then page=page+1;
call symput("tpage",compress(put(page,best.)));
run;

/* Standard - leave this */
options number nodate orientation=landscape papersize=&p_pgsize missing='
' NOQUOTELNMAX/*turn off warnings about quoted strings to long*/;
ods escapechar='`';
%let linetop = \brdrt\brdrs\brdrw30;  * needs to be 1.5pt so calculated
in twips (1/20 pt) ;
%let linebot = \brdrb\brdrs\brdrw30;
%let linebot2 = \brdrb\brdrs\brdrw15;

ods path stdlib.tl06326 (read) ;
ods results off;
ods rtf toc_data/* contents*/
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 &tpage;

ODS PROCLABEL = ' ';
title ;
footnote;
%let wd=0;

data comp;
  set paging end=eof;
  by paramn;
  where page=&i;
  /* Amend title as needed */
  _firtitl="Listing 15.4.4.3.1 Analysis of Pharmacokinetic
Parameters of Nicotine by Bootstrapping Techniques - `n PK Population";
  _upcas=(length(_firtitl)-
length(compress(_firtitl,'ABCDEFGHIJKLMNOPQRSTUVWXYZ')))/2;
  len=&blankn.-length("(Page &i of &tpage)");
  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;
/* Update with your variables as needed */
proc report data = comp headline headskip missing nowd spanrows split =
'#'
%if &i=1 %then %do; contents=' ' %end; %else %do; contents='' %end;;

/*%if &i=1 %then %do;*/
/*column flag page paramn ord stat ("Group-1 PK &linebot." col1 col2) */
/*      ("Group-2 PK &linebot." col3 col4); */
/*      define flag / order noprint;*/
/*      define page      / order order = internal noprint;*/
/*      define paramn     / group style={just=left cellwidth=1cm}
"Variable" format=param.;*/
/*      define ord        / order noprint;*/
/*      define stat       / display style={just=left cellwidth=4cm}
"Statistic";*/
/*      define col1       / display style={just=c cellwidth=1.5cm}
style(header)={just=center} "&col11";*/
/*      define col2       / display style={just=c cellwidth=1.5cm}
style(header)={just=center} "&col12";*/
/*      define col3       / display style={just=c cellwidth=1.5cm}
style(header)={just=center} "&col21";*/
/*      define col4       / display style={just=c cellwidth=1.5cm}
style(header)={just=center} "&col22";*/
/**/
/*      BREAK AFTER ORD / SKIP;*/
/**/
/*%end;*/

/*%else %do; */
column flag page paramn replicate ("Group-1 PK &linebot." ("&col11" col1)
("&col21" col2) ("&col31 &linebot." col3))
      ("Group-2 PK &linebot." ("&col12" col4) ("&col22" col5) ("&col32
&linebot." col6));
      define flag / order noprint;
      define page      / order order = internal noprint;
      define paramn     / group style={just=left cellwidth=1cm} "Variable"
format=param.;
      define replicate   / display style={just=left cellwidth=1cm}
"Replicate";
      define col1       / display style={just=c cellwidth=1.5cm}
style(header)={just=center} "Geometric#LS Mean";
      define col2       / display style={just=c cellwidth=1.5cm}
style(header)={just=center} "Geometric#LS Mean";
      define col3       / display style={just=c cellwidth=1.5cm}
style(header)={just=center} "Estimate";

```

```

        define col4          / display style={just=c cellwidth=1.5cm}
style(header)={just=center} "Geometric#LS Mean";
        define col5          / display style={just=c cellwidth=1.5cm}
style(header)={just=center} "Geometric#LS Mean";
        define col6          / display style={just=c cellwidth=1.5cm}
style(header)={just=center} "Estimate";
/*%end;*/

break after page / page;

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

        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."};
        %IF &I=&TPAGE %THEN %DO;
LINE "Note: &FOOT1; &FOOT2; THS = Tobacco Heating System.";

        line "&seed. is the seed used to generate samples from the SAS
procedure proc survey select";
        line "Note: Estimates are the adjusted geometric least squares
means based on an ANOVA model.";
        line "";
line "Path: &TFLpath.";
%END;
line "Program Run: &sysdate &sysuserid Program Status: &status"
&_blankn.*"\~\~" "(Page &i of &tpage)";
        endcomp;

run;
%end;
/*ods rtf close;*/
ods results on;
ods path reset;

%mend ;

%outrtf(blankn=67, halfblnk=N );

proc printto ; run;

```

```

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

/*****
*****/
/*OLD CODE COMMENTED OUT 2) AMH 08Aug2014 */
/*****
*****/
%MACRO COMMENT;

/*transpose for output*/
proc sort data=listout;
    by analgrln paramn replicate;
run;

proc transpose data=listout out=tlistout(drop=_NAME_) prefix=col;
    by analgrln paramn replicate;
    id colord;
    var out;
run;

/* Standard - macro for paging */
%macro outrtf(blankn=67, halfblnk=N);

/* treatment column headers and footnotes */
/*group 1*/
%let col11=THS 2.2 Menthol (N=&tot11);
%let col21=mCC (N=&tot21);
%let col31=THS 2.2 Menthol:mCC Ratio (%);
%let foot1=%str(mCC = menthol conventional cigarettes);
%let foot11=mCC;

/*group 2*/
%let col12=THS 2.2 Menthol (N=&tot12);
%let col22=NRT gum (N=&tot22);
%let col32=THS 2.2 Menthol:NRT gum Ratio (%);
%let foot2=%str(NRT gum = Nicotine Replacement Therapy gum);
%let foot12=NRT gum;

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

/* Standard - just change the number to match the listing you're working
on. Also change the letters in the*/

```



```

/* bracket, eg ccb = current cigarette brands. Make sure to do this at
the top of the code too. */

%let tflno=L_15_04_04_03_01(PK);

/* 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",""));
run;

/*page numbers*/
data paging;
    set tlistout;
    by analgrln paramn ;
    flag=1;
    retain ln 0 page 0;
    ln+1;
    if first.analgrln or ln>18 then do;
        page+1;
        ln=1;
    end;
    if last.analgrln then call symput("tpage",compress(page,best.));
run;

data _null_;
    set seed;
    call symput('seed',put(seed,8.0));
run;

/* Standard - leave this */
options number nodate orientation=landscape papersize=&p_pgsize missing='
' NOQUOTELNMAX/*turn off warnings about quoted strings to long*/;
ods escapechar='`';
%let linetop = \brdrt\brdrs\brdrw30; * needs to be 1.5pt so calculated
in twips (1/20 pt) ;
%let linebot = \brdrb\brdrs\brdrw30;
%let linebot2 = \brdrb\brdrs\brdrw15;

ods path stdlib.tl06326 (read) ;
ods results off;
ods rtf toc_data/* contents*/
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 &tpage;

```

```

ODS PROCLABEL = ' ';
title ;
footnote;
%let wd=0;

data comp;
    set paging end=eof;
    by paramn;
    where page=&i;
    call symput('grp',compress(put(analgrln,best.)));
    /* Amend title as needed */
    _firtitl="Listing 15.4.4.3.1 Analysis of Pharmacokinetic
Parameters of Nicotine by Bootstrapping Techniques - PK Population";
    call symput('title1',"Table 15.2.4.3.1 Analysis of
Pharmacokinetic Parameters of Nicotine by Bootstrapping Techniques -");
    call symput('title2',"PK Population");
    _upcas=(length(_firtitl)-
length(compress(_firtitl,'ABCDEFGHIJKLMNOPQRSTUVWXYZ')))/2;
    len=&blankn.-length("(Page &i of &tpage)");
    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;
/* Update with your variables as needed */
proc report data = comp headline headskip missing nowd spanrows split =
'#'
%IF &I=1 %THEN %DO; CONTENTS=' ' %END; %ELSE %DO; CONTENTS='' %END;;
    column flag page paramn replicate ("Group-&grp PK &linebot."
("&&col1&grp" col1) ("&&col2&grp" col2) ("&&col3&grp &linebot." col3
col4)) ;
    define flag / order noprint;
    define page / order order = internal noprint;
    define paramn / group style={just=left cellwidth=1cm} "Variable"
format=param.;
    define replicate / display style={just=left cellwidth=1cm}
"Replicate";
    define col1 / display style={just=c cellwidth=1.5cm}
style(header)={just=center} "Geometric LS Mean";
    define col2 / display style={just=c cellwidth=1.5cm}
style(header)={just=center} "Geometric LS Mean";

```

```

        define col3          / display style={just=c cellwidth=1.5cm}
style(header)={just=center} "Estimate";
        define col4          / display style={just=c cellwidth=1.5cm}
style(header)={just=center} "CV%";

        break after page / page;

break before flag / page %IF &I=1 %THEN %DO;
        CONTENTS="&_FSRTITL" %END; %ELSE %DO; CONTENTS='' %END;;

        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 "\b\fs24\sa24 &title1." ;
        line "\b\fs24\sa24 &title2." ;
        line "&linebot";
        endcomp;

        compute after _page_ / style={just=left protectspecialchars=off
pretext="&linetop."};
        line "Note:  &&foot&grp; THS = Tobacco Heating System.";
        line "&seed. is the seed used to generate samples from the SAS
procedure proc survey select";
        line "Note: Estimates are the adjusted geometric least squares
means based on an ANOVA model.";
        line "";
        line "Path: &TFLpath." &_blankn.*"\~\~" "(Page &i of
&tpage)";
        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=67, halfblnk=N );

proc printto ; run;

%MEND; /* 2) AMH 08Aug2014 */
/*****
*****/

```

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