Full Programming Examples

Table of Contents

A Demographic Table (Pre-ODS)

In this example a basic text formatted table is created with an analysis of Age, Gender, Height and Weight.

This uses a number of macros that have sets of code that can be used in other programs. Lets look at the first macro STDSTAT which computes the n, mean, standard deviation, median, minimum and maximum.

```
%macro stdstat(in=      /*Dataset with data*/
               ,vr=     /*Variable*/
               ,ot=     /*Output dataset*/
               ,dp=0    /*Default Decimal Place setting for raw data*/
               );

   *Do statistical analysis by TRTGRP;
   proc summary data=&in nway;
      class trtgrp;
      var &vr;
      output out=&ot.a (drop=_type freq_)
             n=zn mean=zm std=zs median=ze min=zi max=za;
   run;

   *Output result;
   data &ot;
      attrib trtgrp length=8
             statcode length=8
             statname length=$100 informat=$char100. format=$char100.
             statrslt length=$20 informat=$char20. format=$char100.;
      set &ot.a;
      keep trtgrp statcode statname statrslt;
      statcode=1;
       statname='  n';
       statrslt=put(zn,3.);
       output;
      statcode=2;
       statname='  Mean';
       statrslt=put(zm,%eval(5+&dp).%eval(1+&dp));
       output;
      statcode=3;
       statname='  Std. Dev.';
       statrslt=put(zs,%eval(6+&dp).%eval(2+&dp));
       output;
      statcode=4;
       statname='  Median';
       statrslt=put(ze,%eval(5+&dp).%eval(1+&dp));
       output;
      statcode=5;
       statname='  Minimum';
       %if &dp=0 %then statrslt=put(zi,3.);
       %else statrslt=put(zi,%eval(4+&dp).&dp);;
       output;
```

```
      statcode=6;
       statname='  Maximum';
      %if &dp=0 %then statrslt=put(za,3.);
       %else statrslt=put(za,%eval(4+&dp).&dp);;
       output;
   run;
%mend stdstat;
```

The second macro is one that constructs the titles and footnotes. While this is a basic one that could be written has just a set of TITLE and FOOTNOTE statements this does have the advantage of either aligning text left, center and right, all on the one line. What this does need a value for a macro variable called LZ which is the line size for the page of output, which in this case, if you look at the code later, is set by a line

%let lz=%sysfunc(getoption(LINESIZE));

near the start of the program.

```
%macro titlfoot(tv  /*Title or footnote reference*/
               ,tl  /*Text left aligned*/
               ,tc  /*Text center aligned*/
               ,tr  /*Text right aligned*/
               );

  *Note: macro requires the macro variable LZ to be set as the width of
   the page in characters prior to the call of the macro;
  data _null_;

    *Build blank title/footnote line. Blanks in line are being held by
     use of $CHAR format and informat;
    attrib _txt length=$&lz informat=$char&lz.. format=$char&lz..;
    _txt=repeat(' ',%eval(&lz.-1));

    *Left align text;
    _tln=lengthn("&tl");
     if _tln>0 then substr(_txt,1,_tln)="&tl";

    *Center align text;
    _tcn=lengthn("&tc");
    if _tcn>0 then do;
      if (&lz-_tcn) in(1,0) then _tcl=1;
      else _tcl=int((&lz-_tcn)/2);
      substr(_txt,_tcl,_tcn)="&tc";
    end;

    *Right align text;
    _trn=lengthn("&tr");
    if _trn>0 then do;
      _trl=&lz-_trn+1;
      substr(_txt,_trl,_trn)="&tr";
    end;
    call symput("&tv.t",_txt);
  run;

  *Set title/footnote statement;
  &tv "&&&tv.t";
%mend titlfoot;
```

A third macro is use in the generation of this table, %PAGENOOF. In the first title there is a setting of ‘XXXX’ (first line, right aligned) which is then replaced with ‘Page 1 of 2’ etc., I.e. the the page number of the total number of pages. Note that prior to this call any PROC PRINTTO destination must be closed. The macro itself is:

```
%macro pagnoof(fo);

   *Count the number of pages, I.e. the number of ‘XXXX’ strings;
   data _null_;
      retain _pagenum 0;
      infile "&fo" length=len end=eof lrecl=200;
      input _txt $varying200. len;
      if index(_txt,'XXXX') then _pagenum+1;
      if eof then call symput('pgnum',compress(put(_pagenum,8.)));
   run;

   *Read in file and replace ‘XXXX’ with ‘Page x of y’;
   data _null_;
      retain _pagenum 0;
      length _txt2 $18;
      infile "&fo" length=len end=eof lrecl=200 sharebuffers;
      file "&fo" lrecl=200;
      input _txt $varying200. len;
      if index(_txt,'XXXX') then do;
         _pagenum+1;
         _txt2='Page '||compress(put(_pagenum,4.))||" of &pgnum";
         _txt2n=lengthn(_txt2);
         _txt2l=&lz.-_txt2n+1;
         substr(_txt,_txt2l,_txt2n)=strip(_txt2);
         len=lengthn(_txt);
      end;
      put _txt $varying200. len;
   run;
%mend pagnoof;
```

Finally the program runs to create the table:

```
/*-----------------------------------------------------------------------*
  Program Name:   BaseDemo.SAS
  Purpose:        Baseline Demography Statistics

  Input Datasets: Dummy Data - Card File (this is included in this file to make
                  the program self-contained)
  Output Files:   BaseDemo.txt
  Macros Used:    STDSTAT, TITLFOOT, PAGNOOF (these are included in this file
                  to make the program self-contained)

  Notes:          This file and the files it accesses are for review
                  purposes only and may NOT be copied or used in any other way
                  without prior consent from the author.  If ILLEGAL use is
                  made of this code, no responsibility will be taken by the
                  author or agents for use of this code and any loss and/or
                  damage that may result.

  History:        Original, David Franklin, 04-Mar-2003

  (c)MMIII David Franklin
```

```
 *-------------------------------------------------------------------------------*/

*-------------------------------------------------------------------------------*;
* Set runtime options.
*-------------------------------------------------------------------------------*;
options ps=54 ls=80 missing=' ' nodate nonumber mprint mlogic symbolgen; run;
%let lz=%sysfunc(getoption(LINESIZE));

*-------------------------------------------------------------------------------*;
* Create required formats necessary for the table
*-------------------------------------------------------------------------------*;
proc format;
    value sectnof
      1=' '
      2='Gender'
      3='Age (years) (1)'
      4='Height (cm) (2)'
      5='Weight (kg) (3)';
    value gender
      1='  Male'
      2='  Female';
run;

*-------------------------------------------------------------------------------*;
* Data for Table
*-------------------------------------------------------------------------------*;
data demog;
    infile cards;
    input patid $1-3 gender birthdt date9. height weight firstdt date9. trtgrp;
    attrib age length=8 label='Age (years)';
    age=floor((firstdt-birthdt)/365.25);   ** From Analysis Plan;
cards;
001 1 14OCT1965 165 85 15APR2002 1
002 2 27APR1975 170 70 16APR2002 2
003 2 01APR1972 140 65 17APR2002 0
004 1 17DEC1985 180 90 18APR2002 2
005 2 05JUL1968 160 75 19APR2002 1
;
run;
proc datasets library=work nodetails nolist;
    modify demog;
        format birthdt date9.
               firstdt date9.;
        label patid='Patient ID'
              gender='Gender'
              birthdt='Date of Birth'
              height='Height (cm) at Screening'
              weight='Weight (kg) at Baseline'
              firstdt='Date of First Treatment'
              trtgrp='Treatment Group';
    quit;
run;


*-------------------------------------------------------------------------------*;
* Create and All Patients Treatment Group
*-------------------------------------------------------------------------------*;
data demog1;
```

```
    set demog;
    output;
    if trtgrp in(1,2) then do;
        trtgrp=9;
        output;
    end;
run;

*-----------------------------------------------------------------------------*;
* Do statistical calculations
*-----------------------------------------------------------------------------*;

** Get total patient population;
data _null_;
    if 0 then set demog nobs=npats;
    call symput('npats',put(npats,1.));
    stop;
run;


** Frequency Calculations for Patient Number, Gender;
proc freq data=demog1;
    tables trtgrp /out=patk0 noprint;
    tables trtgrp*gender /out=gender0 noprint;
run;

** Statistical Calculations for Age, Height and Weight;
%stdstat(demog1,age,age0);
%stdstat(demog1,height,height0);
%stdstat(demog1,weight,weight0);

** Append the results into one dataset;
data all0;
    retain npats &npats _flg 0;
    set patk0   (in=a)
        gender0 (in=b)
        age0    (in=c)
        height0 (in=d)
        weight0 (in=e);
    keep sectno trtgrp statcode statname statrslt;
    sectno=sum(a*1,b*2,c*3,d*4,e*5);
    if a then do;
        statcode=0;
        statname='Number of Patients';
        statrslt=put(count,3.)||' ('||put(count*100/npats,3.)||')';
    end;
    else if b then do;
        statcode=gender;
        statname=put(gender,gender.);
        statrslt=put(count,3.)||' ('||put(count*100/npats,3.)||')';
    end;
run;

** Transpose results for report;
proc sort data=all0;
    by sectno statcode statname;
run;
proc transpose data=all0 out=all1 prefix=V;
```

```
    by sectno statcode statname;
    id trtgrp;
    var statrslt;
run;

*-----------------------------------------------------------------------------*;
* Generate Report
*-----------------------------------------------------------------------------*;

** Titles and Footnotes;
%titlfoot(title1,%str(David Franklin, Consultant),,XXXX);
%titlfoot(title2,%str(Study: Jelly Bean Treatment Placebo, A and B));
%titlfoot(title4,,Table 1);
%titlfoot(title5,,Baseline Demographics);
footnote1 "%sysfunc(repeat(%str(-),&lz))"; *Build horizontal line;
%titlfoot(footnote2,%nrquote((1) Age = floor((Date of first treatment - Date of
Birth)/365.25)));
%titlfoot(footnote3,(2) Height is taken at Screening.);
%titlfoot(footnote4,(3) Weight is taken at Baseline.);
%titlfoot(footnote6,BaseDemo.sas,,&sysdate:&systime);

** Generate Report;
proc printto print="c:\BaseDemo.txt" new; run;
proc report data=all1 nowindows headline split='|';
    columns sectno statcode statname v0 ('Active Treatment|---' (v1 v2 v9));
    define sectno /group noprint;
    define statcode /group noprint;
    define statname /group ' ' width=32;
    define v0 /display 'Placebo' width=10;
    define v1 /display '  A' width=10;
    define v2 /display '  B' width=10;
    define v9 /display ' Total' width=10;
    compute before sectno;
        line ' ';
        line @1 sectno sectnof.;
    endcomp;
    quit;
run;
proc printto; run;

** Add Page Numbering;
%pagnoof(c:\BaseDemo.txt);
run;

/*End of File*/
```

The output from the program produces a table:

David Franklin, Consultant Page 1 of 1
Study: Jelly Bean Treatment Placebo,A and B

Table 1
Baseline Demographics

		Active Treatment		
	Placebo	A	B	Total
Number of Patients	1 (20)	2 (40)	2 (40)	4 (80)
Gender				
Male		1 (20)	1 (20)	2 (40)
Female	1 (20)	1 (20)	1 (20)	2 (40)
Age (years) (1)				
n	1	2	2	4
Mean	30.0	34.5	21.0	27.8
Std. Dev.		2.12	7.07	8.88
Median	30.0	34.5	21.0	29.5
Minimum	30	33	16	16
Maximum	30	36	26	36
Height (cm) (2)				
n	1	2	2	4
Mean	140.0	162.5	175.0	168.8
Std. Dev.		3.54	7.07	8.54
Median	140.0	162.5	175.0	167.5
Minimum	140	160	170	160
Maximum	140	165	180	180
Weight (kg) (3)				
n	1	2	2	4
Mean	65.0	80.0	80.0	80.0
Std. Dev.		7.07	14.14	9.13
Median	65.0	80.0	80.0	80.0
Minimum	65	75	70	70
Maximum	65	85	90	90

(1) Age = floor((Date of first treatment - Date of Birth)/365.25)
(2) Height is taken at Screening.
(3) Weight is taken at Baseline.

BaseDemo.sas 04MAR03:19:18

A DM and AE Table (ODS-RTF)

In this example a Demographics and Adverse Event table are created using ODS RTF. Data and the PROC TEMPLATE code are given.

First the PROC TEMPLATE:

```
%macro rtf_template;

   ods path work.templat(update)
            sasuser.templat(read)
            sashelp.tmplmst(read);
   ods path show;

   proc template;
       define style STYL_std;
           parent = Styles.Default;
           replace fonts /
               'TitleFont2' = ("Times New Roman", 10pt)
               'TitleFont' = ("Times New Roman", 10pt)
               'StrongFont' = ("Times New Roman", 10pt)
               'EmphasisFont' = ("Times New Roman", 10pt, Bold)
               'FixedEmphasisFont' = ("Times New Roman", 10pt, Bold)
               'FixedStrongFont' = ("Times New Roman", 10pt)
               'FixedHeadingFont' = ("Times New Roman", 10pt)
               'BatchFixedFont' = ("Times New Roman", 10pt)
               'FixedFont' = ("Times New Roman", 10pt)
               'headingEmphasisFont' = ("Times New Roman", 10pt)
               'headingFont' = ("Times New Roman", 10pt)
               'docFont' = ("Times New Roman", 10pt);
           style SystemTitle from TitlesAndFooters /
               protectspecialchars=off
               asis=on
               font=Fonts('TitleFont');
           replace Output from Container /
               frame = VOID
               rules = NONE
               background=_undef_
               frameborder=OFF;
           replace color_list /
               'bg' = cxFFFFFF
               'fg' = cx000000;
           replace colors /
               'headerfgemph' = color_list('fg')
               'headerbgemph' = color_list('bg')
               'headerfgstrong' = color_list('fg')
               'headerbgstrong' = color_list('bg')
               'headerfg' = color_list('fg')
               'headerbg' = color_list('bg')
               'datafgemph' = color_list('fg')
               'databgemph' = color_list('bg')
               'datafgstrong' = color_list('fg')
               'databgstrong' = color_list('bg')
               'datafg' = color_list('fg')
```

```
                'databg' = color_list('bg')
                'batchfg' = color_list('fg')
                'batchbg' = color_list('bg')
                'tableborder' = color_list('fg')
                'tablebg' = color_list('bg')
                'notefg' = color_list('fg')
                'notebg' = color_list('bg')
                'bylinefg' = color_list('fg')
                'bylinebg' = color_list('bg')
                'captionfg' = color_list('fg')
                'captionbg' = color_list('bg')
                'proctitlefg' = color_list('fg')
                'proctitlebg' = color_list('bg')
                'titlefg' = color_list('fg')
                'titlebg' = color_list('bg')
                'systitlefg' = color_list('fg')
                'systitlebg' = color_list('bg')
                'Conentryfg' = color_list('fg')
                'Confolderfg' = color_list('fg')
                'Contitlefg' = color_list('fg')
                'link2' = color_list('fg')
                'link1' = color_list('fg')
                'contentfg' = color_list('fg')
                'contentbg' = color_list('bg')
                'docfg' = color_list('fg')
                'docbg' = color_list('bg');
            replace Body from Document /
                bottommargin = 1.0in
                topmargin = 1.0in
                rightmargin = 1.0in
                leftmargin = 1.0in
                pagebreakhtml = html('PageBreakLine');
            replace SystemFooter from TitlesAndFooters /
                font = Fonts('TitleFont')
                just=left
                cellpadding=0
                cellspacing=0;
            style table from output /
                background=_Undef_
                frame=hsides
                rules=groups
                borderwidth = 1pt
                cellpadding = 0
                cellspacing = 0;
            style Header from header /
               background=_undef_
                frame=below
                rules=rows
                font = fonts('HeadingFont')
                foreground = colors('headerfg')
                background = colors('headerbg');
            style Rowheader from Rowheader /
                rules=rows
                background=_undef_
                frame=below;
        end;
    run;
%mend rtf_template;
```

Now the program itself. This would typically be two programs but for purposes of the example the one program will produce two outputs, one for a Demographics table, the second an Adverse Event table.

As in any good programming style macros are used — this allows us to have a set of code that is written once and used often in multiple programs. The first we will look at is the macro that sets the population counts:

```
*** Population Count Macro;
%macro _Z_popk(ids=     /*Source dataset*/
              ,pop=     /*Population, default is SAFETY*/
              ,ods=pop0 /*Output dataset*/
              );
   %global YA YB1 YB2 YBT YC YZT;
   %let YA=0;
   %let YB1=0;
   %let YB2=0;
   %let YBT=0;
   %let YC=0;
   %let YZT=0;
   data &ods;
      set &ids (in=a)
          &ids (in=b where=(trtgrpc=:'B'))
          &ids (in=c);
      if b then trtgrpc='BT';
      if c then trtgrpc='ZT';
   run;
   proc summary data=&ods nway;
      class trtgrpc;
      where safety='Y';
      output out=&ods (drop=_type_);
   run;
   data _null_;
      set &ods;
      call symput(compress('Y'||strip(trtgrpc)),
                  compress(put(_freq_,best.)));
   run;
%mend _Z_popk;
```

We know the standard columns in the output so we have a common set of code that gets the population.

The next macro is one that will do counts:

```
*** Frequency Count Macro;
%macro _Z_fcnt(ids=         /*Source dataset*/
              ,ivr=         /*Source variable*/
              ,ods=         /*Output dataset*/
              ,pct=Y        /*Percentage needed Y/N*/
              ,grpv=trtgrpc /*Group analysis variable*/
              );
    proc summary data=&ids nway;
        class &grpv &ivr;
        output out=&ods._A (drop=_type_);
    run;
    data &ods._B;
        merge &ods._A (in=a) key_count (in=b rename=(_freq_=_bign_));
        by &grpv;
```

```
        if a;
        length rslt $30;
        select(_freq_);
            when(0,.) rslt='0';
            otherwise
              rslt=compress(put(_freq_,best.));
        end;
        %if &pct=Y %then %do;
           if _freq_>0 and _bign_>0 then
            rslt=strip(rslt)
                 ||' ('||compress(put(_freq_*100/_bign_,8.))||')';
        %end;
    run;
    proc sort data=&ods._B;
        by &ivr;
        format &grpv $4.;
    run;
    proc transpose data=&ods._B out=&ods prefix=V;
        by &ivr;
        id &grpv;
        var rslt;
    run;
%mend _Z_fcnt;
```

The third is a macro that will do the continuous statistics:

```
*** Summary Statistics Macro;
%macro _Z_suma(ids=  /*Source dataset*/
              ,ivr=  /*Source variable*/
              ,ods=  /*Output dataset*/
              ,dp=0  /*Number of decimal places in raw data*/
              );
    proc summary data=&ids nway;
        class trtgrpc;
        var &ivr;
        output out=&ods._A (drop=_type_ _freq_)
               n=x1 mean=x2 std=x3
               median=x4 min=x5 max=x6;
    run;
    data &ods._B;
        set &ods._A;
        array z{6} x1-x6;
        length rslt $30;
        do i=1 to 6;
          select(i);
            when(1) do;
              rslt=trim(left(put(z{i},8.0)));
              output;
              end;
            when(2,4) do;
              rslt=trim(left(put(z{i},8.%eval(&dp+1))));
              output;
              end;
            when(3) do;
              rslt=trim(left(put(z{i},8.%eval(&dp+2))));
              output;
              end;
            when(5,6) do;
```

```
              rslt=trim(left(put(z{i},8.%eval(&dp+0))));
              output;
              end;
          end;
        end;
        format trtgrpc $4.;
        keep trtgrpc i rslt;
    run;
    proc sort data=&ods._B;
        by i;
    run;
    proc transpose data=&ods._B out=&ods prefix=V;
        by i;
        id trtgrpc;
        var rslt;
    run;
%mend _Z_suma;
```

This standard is what we will use to produce any continuous statistics, whether it be AGE in demographics or a laboratory parameter in a lab analysis.

Now the code for the generation of data, manipulation, counts and stats, and finally reporting for DM and AE tables:

```
*------------------------------------------------------------------------*
  PROGRAM NAME: DMandAE.sas
  DESCRIPTION:  Sample program showing code to generate an RTF
                Template, and then using some data, a basic
                Demographics and Adverse Event in RTF format.

                  ------------------------------------------------
                    SPECIAL NOTE: This code is given only for
                    learning purposes (not to be used for any
                    commercial or business activities) and is to
                    be used "as is" with no implied guarantee or
                    warranty.  No responsibility will be assumed
                    for any damage or loss due to use of this
                    code.
                  ------------------------------------------------

Copyright: David Franklin, MMIX
*------------------------------------------------------------------------*;


*-----------------------------------------------
  Set directory location of RTF File
*-----------------------------------------------;
%let location_rtf_file=%str(C:\tmp\);

*-----------------------------------------------;
*Initialize, options template location & ODS Path;
*-----------------------------------------------;
options pageno=1 nodate source2 mergenoby=warn msglevel=i nobyline
        orientation=landscape nonumber symbolgen mprint mlogic
        mprintnest formchar="|----|+|---+=|-/\<>*" nonumber;
goptions reset=all;
ods escapechar '!';
ods path work.styles(write)
```

```
         sasuser.templat(read)
         sashelp.tmplmst(read);
ods path show;
%rtf_template;

*-----------------------------------------------;
*Make data;
*-----------------------------------------------;
data key (label='Treatment/Population Data');
   attrib USUBJID length=$5 label='Unique Subject ID'
          TRTGRPC length=$2 label='Treatment Group'
          SAFETY length=$1 label='Safety Population';
   infile cards dlm='~' dsd;
   input USUBJID $ TRTGRPC $ SAFETY $;
cards;
1.001~A~Y
1.002~A~Y
1.003~B1~Y
1.004~B1~Y
1.005~B2~Y
2.001~~N
2.002~B1~Y
2.003~B2~Y
2.004~C~Y
2.005~C~Y
;
run;

data dm (label='Demographic Data');
   attrib USUBJID length=$5 label='Unique Subject ID'
          SEX length=$1 label='Sex'
          AGECALC length=8 label='Age (years), Calculated';
   infile cards dlm='~' dsd;
   input USUBJID $ SEX $ AGECALC;
cards;
1.001~F~50
1.002~F~33
1.003~F~51
1.004~M~38
1.005~M~48
2.001~M~.
2.002~M~23
2.003~F~65
2.004~M~79
2.005~F~65
;
run;

data ae (label='Adverse Event Data');
   attrib USUBJID length=$5 label='Unique Subject ID'
          PER length=$5 label='Period'
          AETERM length=$70 label='AE Raw Term'
          AEDECOD length=$70 label='AE Standard Term'
          AEBODSYS length=$70 label='AE Body System';
   infile cards dlm='~' dsd;
   input USUBJID $ PER $ AETERM $ AEDECOD $ AEBODSYS $;
cards;
1.001~TE~HYPERMAGNESEMA~Hypermagnesaemia~Metabolism and nutrition disorders
```

```
1.001~TE~ABDOMINAL CRAMPS~Abdominal pain~Gastrointestinal disorders
1.001~TE~NAUSEA~Nausea~Gastrointestinal disorders
1.002~TE~HYPONATREMIA~Hyponatraemia~Metabolism and nutrition disorders
1.002~TE~FATIGUE~Fatigue~General disorders and administration site conditions
1.002~TE~FATIGUE~Fatigue~General disorders and administration site conditions
1.003~TE~FATIGUED~Fatigue~General disorders and administration site conditions
1.003~TE~ANEMIA~Anaemia~Blood and lymphatic system disorders
1.003~TE~NAUSEA~Nausea~Gastrointestinal disorders
1.003~TE~MILD NAUSEA~Nausea~Gastrointestinal disorders
1.004~TE~VOMITING~Vomiting~Gastrointestinal disorders
1.004~TE~ELEVATED CREATININE~Blood creatinine increased~Investigations
1.004~TE~FEVER~Pyrexia~General disorders and administration site conditions
1.005~TE~ANEMIA~Anaemia~Blood and lymphatic system disorders
1.005~TE~ELEVATED CREATININE~Blood creatinine increased~Investigations
2.001~TE~CONSTIPATION~Constipation~Gastrointestinal disorders
2.001~TE~HYPOCALCEMIA~Hypocalcaemia~Metabolism and nutrition disorders
2.001~TE~AST ELEVATION~Aspartate aminotransferase increased~Investigations
2.003~TE~DIARRHEA~Diarrhoea~Gastrointestinal disorders
2.003~TE~NAUSEA~Nausea~Gastrointestinal disorders
2.003~TE~HEART BURN~Dyspepsia~Gastrointestinal disorders
2.003~TE~AZOTEMIA~Azotaemia~Renal and urinary disorders
2.004~TE~AZOTEMIA~Azotaemia~Renal and urinary disorders
2.004~TE~NAUSEA~Nausea~Gastrointestinal disorders
2.004~TE~FATIGUE~Fatigue~General disorders and administration site conditions
2.004~TE~HEARTBURN~Dyspepsia~Gastrointestinal disorders
2.005~TE~DYSPNEA~Dyspnoea~Respiratory, thoracic and mediastinal disorders
2.005~TE~FEVER~Pyrexia~General disorders and administration site conditions
2.005~TE~HYPOPHOSPHATEMIA~Hypophosphataemia~Metabolism and nutrition disorders
;
run;

*------------------------------------------------;
*Formats;
*------------------------------------------------;
proc format;
   value $vsex
      'M'='Male'
      'F'='Female';
   value vstat
      1='N'
      2='Mean'
      3='Std Dev'
      4='Median'
      5='Minimum'
      6='Maximum';
run;

*------------------------------------------------;
*Do calculations;
*------------------------------------------------;

*** Population Count (Safety);
%_Z_popk(ids=key,pop=safety,ods=key_count); *** Population counts;


************************;
*** Demographic data ***;
************************;
```

```
data dm_key;
   merge dm (in=a) key (in=b);
   by usubjid;
   if safety='Y';
run;

data dm_key0;  *** Create a total B and All Total;
   set dm_key (in=a)
       dm_key (in=b where=(trtgrpc=:'B'))
       dm_key (in=c);
   if b then trtgrpc='BT';
   if c then trtgrpc='ZT';
run;

%_Z_fcnt(ids=dm_key0,ivr=sex,ods=dm_rslt1); *** Sex Count;
%_Z_suma(ids=dm_key0,ivr=agecalc,ods=dm_rslt2,dp=0); *** Age Summary;

data dm_rpt0; *** Bring results together;
   set dm_rslt1 (in=a)
       dm_rslt2 (in=b);
   length sectno subsect 8
          label statlbl $70;
   if a then do;
      sectno=1;
      if sex='M' then subsect=1;
      else if sex='F' then subsect=2;
      label='!R"\li180 "'||put(sex,$VSEX.);
      statlbl='N (%)';
      array z{*} $30 V00 VA VB1 VB2 VBT VC VZT;
      do i=1 to dim(z);
        if missing(z(i)) then z(i)='0';
      end;
   end;
   else if b then do;
      sectno=2;
      subsect=i;
      if subsect=1 then label='Age (yrs)';
      statlbl=put(i,vstat.);
   end;
   tmp=' ';
run;

data dm_addtitle; *** Add a blank line for SEX in title;
   set dm_rpt0 (obs=1);
   if _n_=1 then call missing(of _all_);
   statlbl='';
   sectno=1; subsect=0; label='Sex'; output;
run;
proc append base=dm_rpt0 data=dm_addtitle;
run;
proc sort data=dm_rpt0;
  by sectno subsect;
run;

**************;
*** AE data***;
**************;
data ae_key;
```

```
   merge ae (in=a) key (in=b);
   by usubjid;
   if safety='Y';
run;

data ae_key0;  *** Create a total B and All Total;
   set ae_key (in=a)
       ae_key (in=b where=(trtgrpc=:'B'))
       ae_key (in=c);
   if b then trtgrpc='BT';
   if c then trtgrpc='ZT';
run;

proc sort data=ae_key0 (keep=usubjid trtgrpc)
          out=ae_pk1 nodupkey;  *** AE Subject Count;
   by _all_;
run;
data ae_pk1;
   retain all 'ALL';
   set ae_pk1;
run;
%_Z_fcnt(ids=ae_pk1,ivr=all,ods=ae_rslt0);

proc sort data=ae_key0 (keep=usubjid trtgrpc aebodsys)
          out=ae_soc1 nodupkey; *** AE Body System Count;
   by _all_;
run;
%_Z_fcnt(ids=ae_soc1,ivr=aebodsys,ods=ae_rslt1);

proc sort data=ae_key0 (keep=usubjid trtgrpc aebodsys aedecod)
          out=ae_pt1 nodupkey;
   by _all_;
run;
%_Z_fcnt(ids=ae_pt1,ivr=aebodsys aedecod,ods=ae_rslt2);

data ae_rpt0;  *** Bring the data together;
   set ae_rslt2 (in=a) ae_rslt1 (in=b) ae_rslt0 (in=c);
   aesrc=sum(a*1,b*2,c*3);
   if aesrc=3 then aebodsys='AAA';
run;
proc sort data=ae_rpt0;
   by aebodsys aedecod;
run;

proc sort data=ae_rslt1_a out=ae_sortordA (drop=trtgrpc); *** Get SORT Order;
   by aebodsys;
   where trtgrpc='ZT';
run;
proc sort data=ae_rslt2_a out=ae_sortordB (drop=trtgrpc);
   by aebodsys aedecod;
   where trtgrpc='ZT';
run;
data ae_sortordC;
   merge ae_sortordA (rename=(_freq_=X))
         ae_sortordB (rename=(_freq_=Y));
   by aebodsys;
run;
data ae_sortordC;
```

```
   set ae_sortordC ae_sortordA (in=b rename=(_freq_=X));
run;
proc sort data=ae_sortordC;
   by aebodsys aedecod;
run;
data ae_sortordC;
   modify ae_sortordC;
   if ^missing(x) then x=1000-x;
   if ^missing(y) then y=1000-y;
run;

data ae_rpt1;  *** Setup for list;
  merge ae_rpt0 ae_sortordC;
  by aebodsys aedecod;
  length label statlbl $70;
  if aesrc=3 then x=0;
  y=max(0,y);
  if aesrc=3 then label='At Least One Adverse Event';
  else if ^missing(aedecod) then
    label='!R"\li144 "'||trim(aedecod);
  else do;
      label=trim(aebodsys);
      statlbl='N (%)';
  end;
  array z{*} $30 V00 VA VB1 VB2 VBT VC VZT;
  do i=1 to dim(z);
     if missing(z(i)) then z(i)='0';
  end;
  pg=1;
  tmp=' ';
  if aesrc=3 then statlbl='N';
  else statlbl='N (%)';
run;

*------------------------------------------------;
*Generate DM Table;
*------------------------------------------------;

ods listing close;
ods rtf file="&location_rtf_file\RTF_1DM.rtf" style=custom
        nogtitle nogfootnote;

*Titles;
title1 j=l "Cook and Eat (Yum Yum), Inc." j=r "Page !{pageof}";
title2 j=l "New Sauce";
title3 j=c "Table 1";
title4 j=c "Demographic and Baseline Characteristics (Safety Population)";
footnote1 '!R"\brdrt\brdrs\brdrw20 "'
          "Program Name: DM.sas   "
          "Creation Date and Time: &sysdate9:&systime";

%let undrln=%str(!R'\brdrb\brdrs\brdrw10 ');
%let stdsty=%str(style={cellwidth=1.05in just=left
                        protectspecialchars=off
                        pretext='\tqdec\tx750 ' }
                 style(header)={just=center});
proc report data=dm_rpt0 nowindows headline split='|' missing;
   columns sectno subsect label statlbl
```

```
          VA
          ("&undrln.Blue" (VB1 VB2 VBT))
          VC VZT;
   define sectno  /group order=internal noprint;
   define subsect /group order=internal noprint;
   define label   /display 'Parameter'
     style(column)={cellwidth=1.9in just=left}
     style(header)={just=left};
   define statlbl /display 'Statistic'
     style(column)={cellwidth=0.7in just=left}
     style(header)={just=left};
   define VA      /display "Green|(N=&YA.)"
     &stdsty;
   define VB1     /display "Standard|(N=&YB1.)"
     &stdsty;
   define VB2     /display "Royal|(N=&YB2.)"
     &stdsty;
   define VBT     /display "Total|(N=&YBT.)"
     &stdsty;
   define VC      /display "Red|(N=&YC.)"
     &stdsty;
   define VZT     /display "Overall|(N=&YZT.)"
     &stdsty;
   compute after sectno;
      line ' ';
   endcomp;
   quit;
run;
ods rtf close;

*------------------------------------------------;
*Generate AE Table;
*------------------------------------------------;
ods rtf file="&location_rtf_file\REF_1AE.rtf" style=custom
        nogtitle nogfootnote;

*Titles;
title1 j=l "Cook and Eat (Yum Yum), Inc." j=r "Page !{pageof}";
title2 j=l "New Sauce";
title3 j=c "Table 2";
title4 j=c "Treatment Emergent AE Summary (Safety Population)";
footnote1 '!R"\brdrt\brdrs\brdrw20 "'
          "Program Name: AE.sas   "
          "Creation Date and Time: &sysdate9:&systime";

%let undrln=%str(!R'\brdrb\brdrs\brdrw10 ');
%let stdsty=%str(style={cellwidth=0.9in just=left
                      protectspecialchars=off
                       pretext='\tqdec\tx550 ' }
               style(header)={just=center});
proc report data=ae_rpt1 nowindows headline split='|' missing;
  columns x aebodsys y aedecod label statlbl
      VA
      ("&undrln.Blue" (VB1 VB2 VBT))
      VC VZT;
  define x        /group order=internal noprint;
  define aebodsys /group order=internal noprint;
  define y        /group order=internal noprint;
```

```
  define aedecod /group order=internal noprint;
  define label    /display 'System Organ Class|Preferred Term'
    style(column)={cellwidth=2.5in just=left}
    style(header)={just=left};
  define statlbl /display 'Statistic'
    style(column)={cellwidth=0.75in just=left}
    style(header)={just=left};
 define VA      /display "Green|(N=&YA.)"
   &stdsty;
 define VB1     /display "Standard|(N=&YB1.)"
   &stdsty;
 define VB2     /display "Royal|(N=&YB2.)"
   &stdsty;
 define VBT     /display "Total|(N=&YBT.)"
   &stdsty;
 define VC      /display "Red|(N=&YC.)"
   &stdsty;
 define VZT     /display "Overall|(N=&YZT.)"
   &stdsty;
  compute after aebodsys;
      line ' ';
  endcomp;
  quit;
run;

ods rtf close; ods listing; options number;

/*EOF*/
```

The output from this program for the Demographics table is:

Cook and Eat (Yum Yum), Inc. Page 1 of 1
New Sauce

Table 1
Demographic and Baseline Characteristics (Safety Population)

			Blue				
Parameter	Statistic	Green (N=2)	Standard (N=3)	Royal (N=2)	Total (N=5)	Red (N=2)	Overall (N=9)
Sex							
Male	N (%)	0	2 (67)	1 (50)	3 (60)	1 (50)	4 (44)
Female	N (%)	2 (100)	1 (33)	1 (50)	2 (40)	1 (50)	5 (56)
Age (yrs)	N	2	3	2	5	2	9
	Mean	41.5	37.3	56.5	45.0	72.0	50.2
	Std Dev	12.02	14.01	12.02	15.64	9.90	17.53
	Median	41.5	38.0	56.5	48.0	72.0	50.0
	Minimum	33	23	48	23	65	23
	Maximum	50	51	65	65	79	79

Program Name: DM.sas Creation Date and Time: 14JUN2009:10:59

and for the AE table there are two pages:

Table 2
Treatment Emergent AE Summary (Safety Population)

System Organ Class Preferred Term	Statistic	Green (N=2)	Blue Standard (N=3)	Blue Royal (N=2)	Blue Total (N=5)	Red (N=2)	Overall (N=9)
At Least One Adverse Event	N	2 (100)	3 (100)	2 (100)	5 (100)	2 (100)	9 (100)
Gastrointestinal disorders	N (%)	1 (50)	2 (67)	1 (50)	3 (60)	1 (50)	5 (56)
Nausea	N (%)	1 (50)	1 (33)	1 (50)	2 (40)	1 (50)	4 (44)
Dyspepsia	N (%)	0	0	1 (50)	1 (20)	1 (50)	2 (22)
Abdominal pain	N (%)	1 (50)	0	0	0	0	1 (11)
Diarrhoea	N (%)	0	0	1 (50)	1 (20)	0	1 (11)
Vomiting	N (%)	0	1 (33)	0	1 (20)	0	1 (11)
General disorders and administration site conditions	N (%)	1 (50)	2 (67)	0	2 (40)	2 (100)	5 (56)
Fatigue	N (%)	1 (50)	1 (33)	0	1 (20)	1 (50)	3 (33)
Pyrexia	N (%)	0	1 (33)	0	1 (20)	1 (50)	2 (22)
Metabolism and nutrition disorders	N (%)	2 (100)	0	0	0	1 (50)	3 (33)
Hypermagnesaemia	N (%)	1 (50)	0	0	0	0	1 (11)
Hyponatraemia	N (%)	1 (50)	0	0	0	0	1 (11)
Hypophosphataemia	N (%)	0	0	0	0	1 (50)	1 (11)
Blood and lymphatic system disorders	N (%)	0	1 (33)	1 (50)	2 (40)	0	2 (22)
Anaemia	N (%)	0	1 (33)	1 (50)	2 (40)	0	2 (22)
Investigations	N (%)	0	1 (33)	1 (50)	2 (40)	0	2 (22)
Blood creatinine increased	N (%)	0	1 (33)	1 (50)	2 (40)	0	2 (22)
Renal and urinary disorders	N (%)	0	0	1 (50)	1 (20)	1 (50)	2 (22)
Azotaemia	N (%)	0	0	1 (50)	1 (20)	1 (50)	2 (22)

Table 2
Treatment Emergent AE Summary (Safety Population)

System Organ Class Preferred Term	Statistic	Green (N=2)	Blue Standard (N=3)	Blue Royal (N=2)	Blue Total (N=5)	Red (N=2)	Overall (N=9)
Respiratory, thoracic and mediastinal disorders	N (%)	0	0	0	0	1 (50)	1 (11)
Dyspnoea	N (%)	0	0	0	0	1 (50)	1 (11)

In the output these are RTF files. Note that these use Times New Roman 10pt font which is allowable for output to the FDA, but you cannot go lower than 8pt and fonts available are Times New Roman, Arial and Courier New.

/*EOF*/