Paper 158-30

Automation of Clinical Trial Laboratory Data Acquisition Using the SAS® System and DDE

Balakrishna Dandamudi, SFBC International

ABSTRACT:

Clinical SAS programmers receive clinical trial laboratory data from external sources, such as centralized laboratories, in non-SAS file formats e.g., Excel workbooks. Before beginning production data transfer, one needs to complete a test transfer to verify the completeness and accuracy of the data transfer. Once the test transfer is successful, the transfer of production data begins. There may be one transfer or multiple transfers, depending on the clinical trial.

This paper discusses an approach in building and using specialized SAS program modules in order to automate the process of reading clinical laboratory data from an Excel workbook into SAS using SAS/BASE and DDE. This approach minimizes the manual intervention.

In particular, this paper will detail a program that can readily be modified to create laboratory data sets, normal range data sets and formats catalog, based on the specifications and data provided in an Excel workbook with multiple data sets found in multiple Excel sheets.

DDE is a viable option for importing or exporting data to or from MS Excel for those users who prefer to use SAS/BASE only.

INTRODUCTION:

At SFBC New Drug Services, we receive laboratory data from our central facility located in Miami, Florida, as well as from many other external laboratories. Typically, we receive laboratory data as an ASCII file, which is our standard format. Sometimes laboratories send us data in an Excel workbook.

The procedure we have implemented in order to accurately and efficiently process Excel workbook laboratory data is presented here.

DDE, CALL EXECUTE, and PROC SQL

Dynamic Data Exchange (DDE) is a method of accessing data from other windows applications. DDE uses a client/server environment. Both applications, Server and Client, must be running and the data that are to be exchanged must be active in both applications. For additional information, refer to SAS Institute Inc. TS325: The SAS System and DDE.

Call execute is a DATA step interface with the macro facility. Call execute is not part of the macro processor but rather a DATA step interface which interacts with the macro facility during execution.

For additional information refer Riba, S. David. SELF-MODIFYING SAS® PROGRAMS: A DATA STEP INTERFACE.

Procedure SQL is the SAS® version of SQL that can be used to define, access, manipulate and query the data. To better understand, Procedure SQL, consider reading the NESUG 2004 Conference paper Clinical Trial Data Validation: Using SAS® PROC SQL Effectively.

This paper anticipates its readers to have a good understanding of the DDE method, CALL EXECUTE DATA step interface, and Procedure SQL. If you are not familiar with any of these, the previously mentioned references and those listed in the reference section provide additional comprehensive information.

LAYOUT

The process of automating laboratory data acquisition is a one-time process. This needs to be taken care of at the time of the test transfer. Once the test transfer is successful, the subsequent transfers are repetitive in nature.

ASSUMPTIONS

The Excel workbook content needs to be standardized. The standards may vary from study to study but need to be consistent throughout each study. In the example used here, the following assumptions are made:

1. The Excel workbook has five work sheets.
2. The first sheet contains data definitions of LABDATA data set.
3. The second sheet contains data definitions of NRMRANGE data set.
4. The third sheet contains the code list (formats) data.

5. The fourth sheet contains the lab data.
6. The fifth sheet contains the normal ranges data.
7. In all the sheets, the data starts on the third row. Rows one and two contain the header.
8. The folder c:\temp exists.

PROCEDURAL STEPS INVOLVED

The following procedural steps are involved in the process of automation of laboratory data acquisition:

1. Initialize Macro variables and Assign LIBREFs.
2. Launch the Excel application.
3. Make a copy of the Excel workbook.
4. Break the program execution dynamically.
5. Collect the Excel workbook attributes.
6. Create a SAS data set from each worksheet
7. Close and delete the Excel workbook.
8. Create formats catalog.
9. Create LABDATA data set.
10. Create NRMRANGE data set.
11. Generate listings and contents.

STEP 1: ASSIGN MACRO VARIABLES AND LIBREFS

Initialize all macro variables related to the Excel file path, Excel file name, output file path and formats catalog location. Assign library references to data sets output folder and formats catalog output folder.

```
%let xlfpath = C:\sugi30\posters;
%let xlfname = Labdata.xls;
%let ofpath  = C:\sugi30\posters;
%let fmtpath = C:\sugi30\posters;
libname outds "&ofpath";
libname library "&fmtpath";
```

STEP 2: LAUNCH THE EXCEL APPLICATION

Since DDE uses the client/server environment, both SAS and Excel need to be invoked. With SAS already running, invoke Excel. To launch the Excel application, use the Roper method to invoke an Excel application in SAS.

```
options noxsync noxwait xmin;
filename xl2sas dde 'excel|system';
data _null_;
      length fid rc stop time 8;
      fid=fopen('xl2sas','s');
      if (fid le 0) then
            do;
                  call system('start excel');
                  stop=datetime()+10;
                  do while (fid le 0);
                        fid=fopen('xl2sas','s');
                        time=datetime();
                        if (time ge stop) then fid=1;
                  end;
            end;
      rc=fclose(fid);
run;
```

STEP 3: MAKE A COPY OF THE EXCEL WORKBOOK

Keep the original file intact and save it in a temporary location with a standard file name. Refer to the standard file name in the program for further processing. Make sure the folder c:\temp exists.

```
data _null_;
      file xl2sas;
      put '[file.close("c:\temp\temp.xls")]';
      put "[open(""&xlfpath" '\' "&xlfname"")]";
run;
data _null_;
      file xl2sas;
      put "[open(""&xlfpath" '\' "&xlfname"")]";
      put '[save.as("c:\temp\temp.xls",1)]';
run;
```

STEP 4: BREAK PROGRAM EXECUTION DYNAMICALLY

```
data _null_;
window confirm color=green rows=18 columns=40 irow=1 icolumn=1
group=initial
#12 @56 "                      on &sysday, &sysdate..                "    color=magenta
#16 @15 '   SUGI 30                                                  '    color=yellow
#17 @15 '   laboratory Data:Acquisition using the SAS System and DDE'    color=yellow
#18 @15 '   ____________________________________________________     '    color=red
#20 @15 '   The Excel file is launched and saved in C:\temp\temp.xls'    color=yellow
#22 @15 '   Make necessary changes to this workbook:                 '    color=yellow
#24 @15 '    1]  Delete sheets not to be processed                   '    color=yellow
#25 @15 '    2]  Delete data not to be processed (rows and columns)  '    color=yellow
#26 @15 '        in each sheet                                       '    color=yellow
#28 @18       "PRESS" color=white
    @25       "[ENTER]" attr=underline color=yellow
    @34       "to continue program execution." color=white
#30 @18       "PRESS [" color=white
    @25       "BREAK" attr=underline color=yellow
    @30       "] icon on Application Tool Bar," color=white
    @61 ' to interrupt the program execution.                       '   color=white
;
display confirm;
stop;
run;
```

The above DATA _NULL_ step creates and displays a window named CONFIRM. The CONFIRM window fills the entire screen. Although this window example does not require any input, ENTER must be pressed to cause SAS execution to proceed to the STOP statement. If the STOP statement is omitted, the DATA step executes endlessly unless END is entered on the command line of the window.

Note: If there are no observations to read, SAS will fail to detect an end-of-file that is necessary to make the DATA step execution stop.

Using the NOINPUT option in the DISPLAY statement makes the window display to appear then quickly disappear.

Utilize the advantage of this DATA _NULL_ step to break the program after saving the Excel workbook as temp.xls. Review the Excel workbook and make necessary modifications to the Excel workbook. Once the Excel workbook is in the predefined standard form, go to the CONFIRM window and press ENTER to cause SAS execution.

STEP 5: COLLECT THE EXCEL WORKBOOK ATTRIBUTES

To create SAS data sets from an Excel workbook, the number of Excel work sheets, their names, and the number of rows and columns to be read in each sheet must be known. Koen Vyverman has researched this extensively in the past. His tireless efforts that have contributed greatly in supporting DDE usage in SAS are sincerely appreciated.

Using his macro loadnames with custom changes, a data set named SHEETNM is created with all the information related to the Excel workbook.

```
data _null_;
  file xl2sas;
  put '[workbook.next()]';
  put '[workbook.insert(3)]';
  put '[workbook.move(,,1)]';
run;

%macro loadnames;
  %local sh wn nsheets;
  %let sh=0;
  %let wn=0;
  %let nsheets=0;

  filename xlmacro dde 'excel|macro1!r1c1:r1000c1' notab lrecl=200;
  data _null_;
    file xlmacro;
    put '=set.value($b$1,get.workbook(4))';
    put '=halt(true)';
    put '!dde_flush';
    file xl2sas;
    put '[run("macro1!r1c1")]';
    put '[error(false)]';
  run;

  filename nsheets dde "excel|macro1!r1c2:r1c2" notab lrecl=200;
  data _null_;
    length nsheets 8;
    infile nsheets;
    input nsheets;
    call symput('nsheets',trim(left(put(nsheets,2.))));
  run;

  %let nsheets=%eval(&nsheets-1);
  data _null_;
    file xl2sas;
    put '[workbook.activate("macro1")]';
    put '[select("r1c1:r100c1,r1c2:r1c2")]';
    put '[clear(1)]';
    put '[select("r1c1")]';
  run;

  data _null_;
    length maccmd $200;
    file xlmacro;
    %do sh=1 %to &nsheets;
      maccmd="=select(!$b$&sh,!$b$&sh)";
      put maccmd;
      put '=set.name("cell",selection())';
      %do wn=1 %to &sh;
        put '=workbook.next()';
        %end;
      put '=set.value(cell,get.workbook(3))';
      put '=workbook.activate("Macro1")';
      %end;
    %do sh=1 %to &nsheets;
      maccmd="=select(!$c$&sh,!$c$&sh)";
      put maccmd;
      put '=set.name("rows",selection())';
      %do wn=1 %to &sh;
        put '=workbook.next()';
```

```
        %end;
        put '=set.value(rows,get.document(10))';
      put '=workbook.activate("Macro1")';
      %end;
    %do sh=1 %to &nsheets;
      maccmd="=select(!$d$&sh,!$d$&sh)";
      put maccmd;
      put '=set.name("cols",selection())';
      %do wn=1 %to &sh;
        put '=workbook.next()';
        %end;
        put '=set.value(cols,get.document(12))';
      put '=workbook.activate("Macro1")';
      %end;
    put '=halt(true)';
    put '!dde_flush';
    file xl2sas;
    put '[run("macro1!r1c1")]';
    put '[error(false)]';
  run;
 filename sheets dde "excel|macro1!r1c2:r&nsheets.c4" lrecl=200;

 data sheetnm;
    length booknm $100 sheetnm $100 nrows 8 ncols 8 dsname $8;
    infile sheets dsd delimiter='09'x notab TRUNCOVER;
    input sheetnm $ nrows $ ncols $;
      booknm=scan(substr(sheetnm,2),1,'.xls');
      sheetnm =reverse(scan(reverse(sheetnm),1,']'));
      dsname = 'DSN' ||compress(put(_N_,best8.));
  run;  filename nsheets clear;
  filename sheets clear;
  filename xlmacro clear;
%mend loadnames;

%loadnames;
```

Note: LRECL= option specifies the logical record length of the output file. If the LRECL= option is omitted, in windows environment, SAS chooses 256 as a default value. Any record larger than LRECL is truncated. In this **loadnames** macro LRECL is assigned as 200. Looking at the log for data truncation and reassigning the value at the higher side of the max can monitor the LRCEL.

STEP 6: CREATE DATA SETS FROM EACH WORKSHEET

The next step is to convert the Excel workbook data to SAS data sets. We have Excel workbook information is in the SHEETNM data set. Using CALL EXECUTE statements in a DATA _NULL_ data step, all Excel workbooks are converted to SAS data sets.

```
data _null_;
      set sheetnm;
      call execute("filename xlfile DDE 'EXCEL|"||sheetnm||
            "!R1C1:R"|| compress(put(nrows,best8.))|| "C"||compress(put(ncols,best8.))||
            "' NOTAB LRECL = 500;");
      call execute("data "||compbl(dsname)||";");
      call execute("      length DSN $10 C1-C"||compress(put(ncols,best8.))||" $200;");
      call execute("  infile xlfile DLM='09'x DSD TRUNCOVER;");
      call execute("  input       C1 - C"||compress(put(ncols,best8.))||" $;");
      call execute("  DSN='"||compbl(dsname)||"';");
      call execute("      if C1 eq '' and C2 eq '' and C3 eq '' and C4 = '' then delete;");
      call execute("      if _N_ in (1,2) then delete;");
      call execute("  run ;");
      call execute("  proc sort data="||trim(dsname)||";by DSN;run;");
run ;
```

```
proc print data=sheetnm;
      title 'Listing of sheet details dataset';
run;
title;
```

Below is another method of using CALL EXECUTE statements in a DATA _NULL_ data step. All of the data sets generated from the Excel workbook for the data review are printed.

```
data _null_;
      set sheetnm;
      call execute(" options date pageno = 1 center orientation = landscape
                        skip=2 ps=55 ls=175 ;");
      call execute(" proc print data="||compress(dsname)|| " width=minimum;");
      call execute("      title 'Listing of Dataset '"||compress(dsname)||"''';");
      call execute(" run;");
run;
title;
```

STEP 7: CLOSE AND DELETE THE EXCEL WORKBOOK

Excel workbook is converted to a SAS data set. The unwanted Excel workbook should be deleted.

```
data _null_;
  file xl2sas;
  put '[file.close("c:\temp\temp.xls")]';
  put '[file.delete("c:\temp\temp.xls")]';
  put '[quit()]';
run;
```

STEP 8: CREATE FORMATS CATALOG

The data corresponding to the formats catalog is in the DSN3 data set. Create the format data set by mapping each variable to the related variable in a format catalog. For both numeric and character formats, the data set must contain the variables FMTNAME, START, and LABEL and use this data set as an input control data set with the CNTLIN= option in the PROC FORMAT statement. The FORMAT procedure uses the data in the input control data set to construct formats. CNTLOUT=option in the PROC FORMAT creates a SAS data set that stores information about formats that are contained in the catalog specified in the LIBRARY= option. Use FMTLIB option to print information about all the formats in the catalog that are specified in the LIBRARY= option.

```
data format;
      set dsn3;
    length  FMTNAME $8 START $8 END $8 LABEL $50;
      retain FMTNAME;
      START = C2;
      END = C3;
      LABEL = C4;
      if C1  = '' then FMTNAME = FMTNAME;
      if C1 ^= '' then FMTNAME = C1;
run;

proc format library = library cntlin = format cntlout = library.formats fmtlib;
run;
```

STEP 9: CREATE LABDATA DATA SET

The data set DSN1 has the data definitions of LABDATA data set using this information first we are going to create a SAS dataset LABDATA shell then we insert data in to it.

```
data _null_;
      set dsn1;
      by DSN;
      if first.dsn then do;
            call execute (" data outds.labdata(label='Laboratory Data');");
            call execute ("   attrib " ||trim(C1)||" length = $" ||trim(C3)
                  ||" format = "||trim(C4)||" label='" ||trim(C5)||"'");
      end;
      if not first.dsn and not LAST.DSN and upcase(trim(C2)) = 'NUM' then do;
            call execute ("   " ||trim(C1)||" length = " ||trim(C3)||
                    " format = "||trim(C4)||" label='" ||trim(C5)||"'");
      end;
      if not first.dsn and not LAST.DSN and upcase(trim(C2)) = 'CHAR' then do;
            call execute ("   " ||trim(C1)||" length = $" ||trim(C3)||
                    " format = "||trim(C4)||" label='" ||trim(C5)||"'");
      end;
      if LAST.DSN then do;
            call execute ("   " ||trim(C1)||" length = $" ||trim(C3)||
                    " format = "||trim(C4)||" label='" ||trim(C5)||"';");
            call execute ("run;");
      end;
run;

data _null_;
      set dsn4;
            call execute (" proc sql;");
            call execute (" insert into outds.labdata (PROTCLID,INVEST, SUBJID,
                           SUBJINIT, SEX, DOB, DATE_COL, AGE, TIMEPT, GROUP,
                           TEST, RESULT, UNITS, ABNORMAL, REPEAT)");
            call execute (" values ('"||trim(C1)||"','"||trim(C2)||"','"
                           ||trim(C3)||"','"||trim(C4)||"',"
                           ||input(trim(C5),BEST8.)||","
                           ||input(trim(C6),mmddyy10.)||","
                           ||input(trim(C7),mmddyy10.)||","
                           ||trim(C8)||",'"||trim(C9)||"','"
                           ||trim(C10)||"','"||trim(C11)||"','"
                           ||trim(C12)||"','"||trim(C13)||"','"
                           ||trim(C14)||"','"||trim(C15)||"')");
            call execute (" ;");
            call execute (" quit;");
run;

proc sql;
      delete from outds.labdata
      where       PROTCLID = '' and INVEST = '' and SUBJID = ''  and SUBJINIT = ''
            and   SEX = . and DOB = . and DATE_COL = .   and AGE = .
            and   TIMEPT  = '' and GROUP = '' and TEST = '' and RESULT = ''
            and   UNITS   = '' and ABNORMAL = '' and REPEAT   = ''
;
quit;
```

STEP 10: CREATE NRMRANGE DATA SET

The data set DSN2 has the data definitions of NRMRANGE data set using this information first create a SAS dataset NRMRANGE shell then insert data in to it.

The method adopted is same as the one in step 9.

```
data _null_;
      set dsn2;
      by DSN;
      if first.dsn then do;
            call execute (" data outds.nrmrange(label='Normal Ranges Data');");
            call execute ("   attrib " ||trim(C1)||" length = $" ||trim(C3)
                        ||"format = " ||trim(C4)||" label='" ||trim(C5)||"'");
      end;
      if not first.dsn and not LAST.DSN and upcase(trim(C2)) = 'NUM' then do;
            call execute ("   " ||trim(C1)||" length = " ||trim(C3)
                       ||" format = "||trim(C4)||" label='" ||trim(C5)||"'");
      end;
      if not first.dsn and not LAST.DSN and upcase(trim(C2)) = 'CHAR' then do;
            call execute ("   " ||trim(C1)||" length = $" ||trim(C3)
                       ||" format = "||trim(C4)||" label='" ||trim(C5)||"'");
      end;
      if LAST.DSN then do;
            call execute ("   " ||trim(C1)||" length = $" ||trim(C3)
                       ||" format = "||trim(C4)||" label='" ||trim(C5)||"';");
            call execute ("run;");
      end;
run;

data _null_;
      set dsn5;
            call execute (" proc sql;");
            call execute (" insert into outds.nrmrange (PROTCLID, INVEST, GROUP,
                           TEST, TESTNAME, SEX, LOAGE, HIAGE, LO, HI,UNITS)");
            call execute (" values ('"||trim(C1)||"','"||trim(C2)||"','"
                                      ||trim(C3)||"','"||trim(C4)||"','"
                                      ||trim(C5)||"',"||input(trim(C6),best8.)
                                      ||","||input(trim(C7),best8.)||","
                                      ||input(trim(C8),best8.)||",'"
                                      ||trim(C9)||"','"||trim(C10)||"','"
                                      ||trim(C11)||"')");
            call execute (" ;");
            call execute (" quit;");
run;

proc sql;
      delete from outds.nrmrange
      where       PROTCLID = '' and INVEST = '' and GROUP = '' and TEST = ''
            and   TESTNAME = '' and SEX = .     and LOAGE = .  and HIAGE = .
            and   LO = ''       and HI = ''     and UNITS = ''
      ;
quit;
```

STEP 11: LISTINGS AND CONTENTS

Generate listings and contents for data review.

```
options date pageno=1 number=1 center orientation=portrait skip=2 ps=77 ls=138;
proc print data=outds.labdata width=minimum;
proc contents data=outds.labdata varnum;
proc print data=outds.nrmrange width=minimum;
proc contents data=outds.nrmrange varnum;run;
```

CONCLUSION

Laboratory data acquisition can be handled in many ways. The procedure developed using DDE, Base/SAS and CALL EXECUTE is an excellent, efficient and accurate tool. Using this application multiple Excel work sheets can be used to generate multiple data sets. DDE is a viable option for importing or exporting data to or from MS Excel for those users who prefer to use SAS/BASE only.

All code developed under Windows 2000 Pro system, running release 8.2 of the SAS System and MS Office2000.

REFERENCES

1. Roper, C. A. “Intelligently Launching Microsoft Excel from SAS, using SCL functions ported to Base SAS”. Proceedings of the Twenty-Fifth Annual SAS Users Group International Conference, paper 97, 2000.

2. SAS Institute, “Technical Support Document #325 – The SAS System and DDE”. http://ftp.sas.com/techsup/download/technote/ts325.pdf, updated 1999.

3. Vyverman, K. “Creating Custom Excel Workbooks from Base SAS® with Dynamic Data Exchange: A Complete Walkthrough”. Proceedings of the Twenty-Seventh Annual SAS Users Group International Conference, paper 27, 2002.

4. Riba, S. David. SELF-MODIFYING SAS® PROGRAMS: A DATA STEP INTERFACE. Observations Technical Journal for SAS Software Users http://support.sas.com/documentation/periodicals/obs/obswww03/toc.html

5. Balakrishna Dandamudi CALL EXECUTE: the More Powerful but Less Well Known SAS® System Routine http://www.philasug.com/PS0403-3.pdf

6. Balakrishna Dandamudi Clinical Trial Data Validation: Using SAS PROC SQL Effectively Proceedings of the NESUG 2004 Conference http://www.nesug.org/html/Proceedings/nesug04/hw/hw06.pdf

ACKNOWLEDGMENTS

The author would like to express his deepest gratitude to Mr. Glenn Adams, Vice President Data Management of SFBC International’s Clinical Pharmacology Services Division and Mr. Stephen Dorian, Director Data Management of SFBC International’s Clinical Pharmacology Services Division. Without their encouragement, this paper might not have come alive.

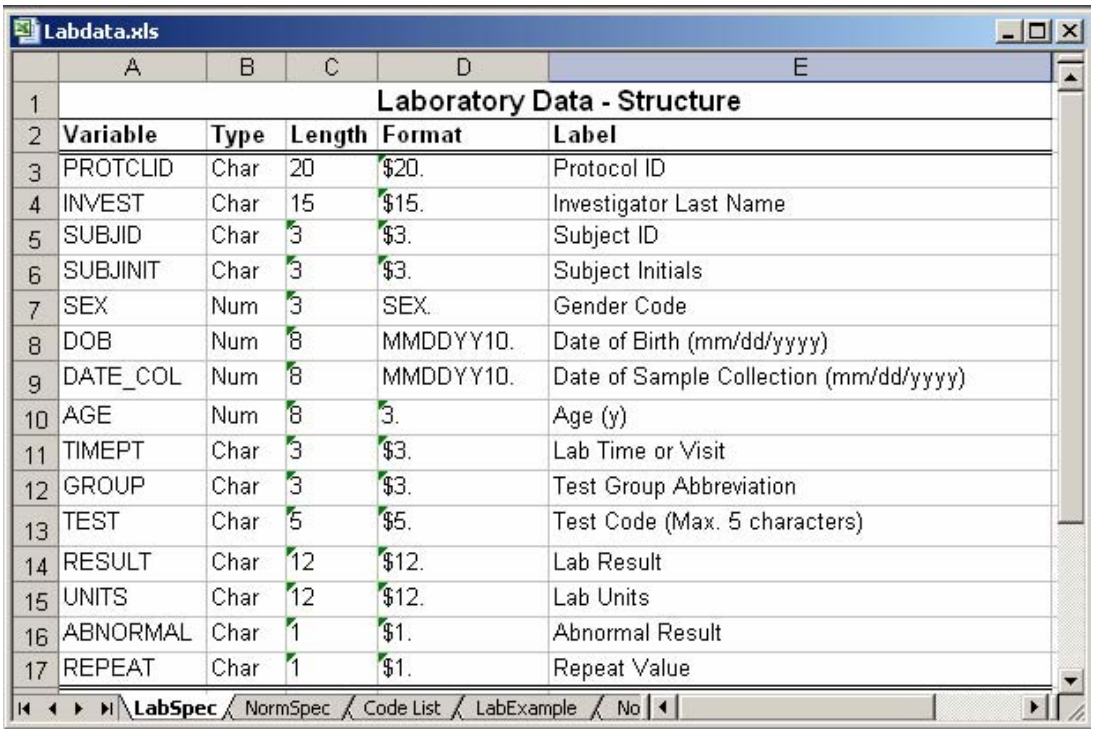
SFBC International, Inc. was founded in 1984 in Miami, Florida as a Phase I clinic, and through organic growth and strategic acquisitions has developed into a world-class contract research organization. SFBC International is now a full service CRO and one of the leading Phase I /II providers in North America. SFBC International has been named by Forbes Magazine as one of the 200 best companies in America for 2002.

TRADEMARKS

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at

Balakrishna Dandamudi
Senior SAS Programmer
SFBC International
Clinical Pharmacology Services
Longwood Corporate Center South
415 McFarlan Road, Suite 201
Kennett Square, PA 19348-2412
Work Phone: 610.444.4722
Fax: 610.444.4663
E-mail: kdandamudi@sfbci.com

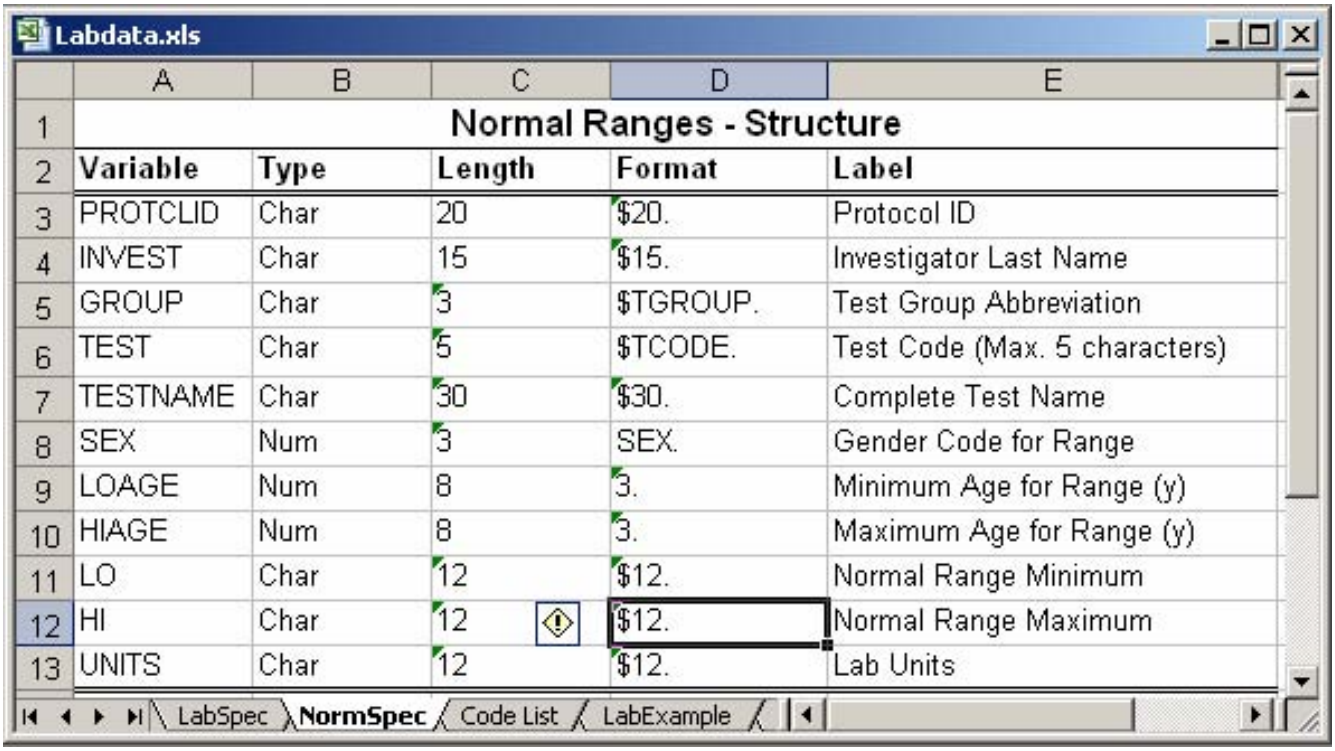


Labdata.xls

	A	B	C	D	E
1	Laboratory Data - Structure				
2	**Variable**	**Type**	**Length**	**Format**	**Label**
3	PROTCLID	Char	20	$20.	Protocol ID
4	INVEST	Char	15	$15.	Investigator Last Name
5	SUBJID	Char	3	$3.	Subject ID
6	SUBJINIT	Char	3	$3.	Subject Initials
7	SEX	Num	3	SEX.	Gender Code
8	DOB	Num	8	MMDDYY10.	Date of Birth (mm/dd/yyyy)
9	DATE_COL	Num	8	MMDDYY10.	Date of Sample Collection (mm/dd/yyyy)
10	AGE	Num	8	3.	Age (y)
11	TIMEPT	Char	3	$3.	Lab Time or Visit
12	GROUP	Char	3	$3.	Test Group Abbreviation
13	TEST	Char	5	$5.	Test Code (Max. 5 characters)
14	RESULT	Char	12	$12.	Lab Result
15	UNITS	Char	12	$12.	Lab Units
16	ABNORMAL	Char	1	$1.	Abnormal Result
17	REPEAT	Char	1	$1.	Repeat Value

LabSpec | NormSpec | Code List | LabExample | No

Fig 1: Excel work sheet 1

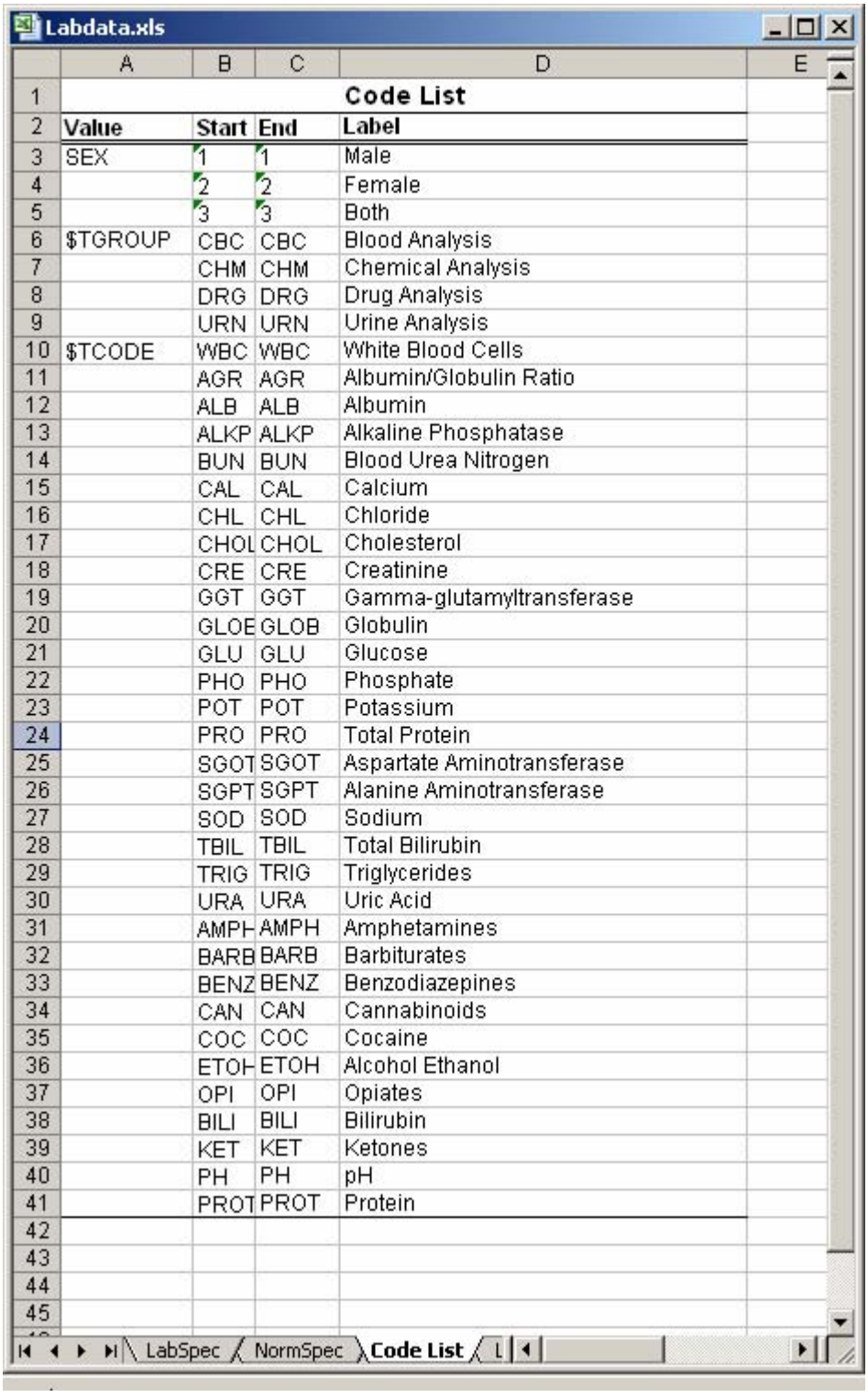


Labdata.xls

	A	B	C	D	E
1	Normal Ranges - Structure				
2	**Variable**	**Type**	**Length**	**Format**	**Label**
3	PROTCLID	Char	20	$20.	Protocol ID
4	INVEST	Char	15	$15.	Investigator Last Name
5	GROUP	Char	3	$TGROUP.	Test Group Abbreviation
6	TEST	Char	5	$TCODE.	Test Code (Max. 5 characters)
7	TESTNAME	Char	30	$30.	Complete Test Name
8	SEX	Num	3	SEX.	Gender Code for Range
9	LOAGE	Num	8	3.	Minimum Age for Range (y)
10	HIAGE	Num	8	3.	Maximum Age for Range (y)
11	LO	Char	12	$12.	Normal Range Minimum
12	HI	Char	12	$12.	Normal Range Maximum
13	UNITS	Char	12	$12.	Lab Units

LabSpec | NormSpec | Code List | LabExample

Fig 2: Excel work sheet 2

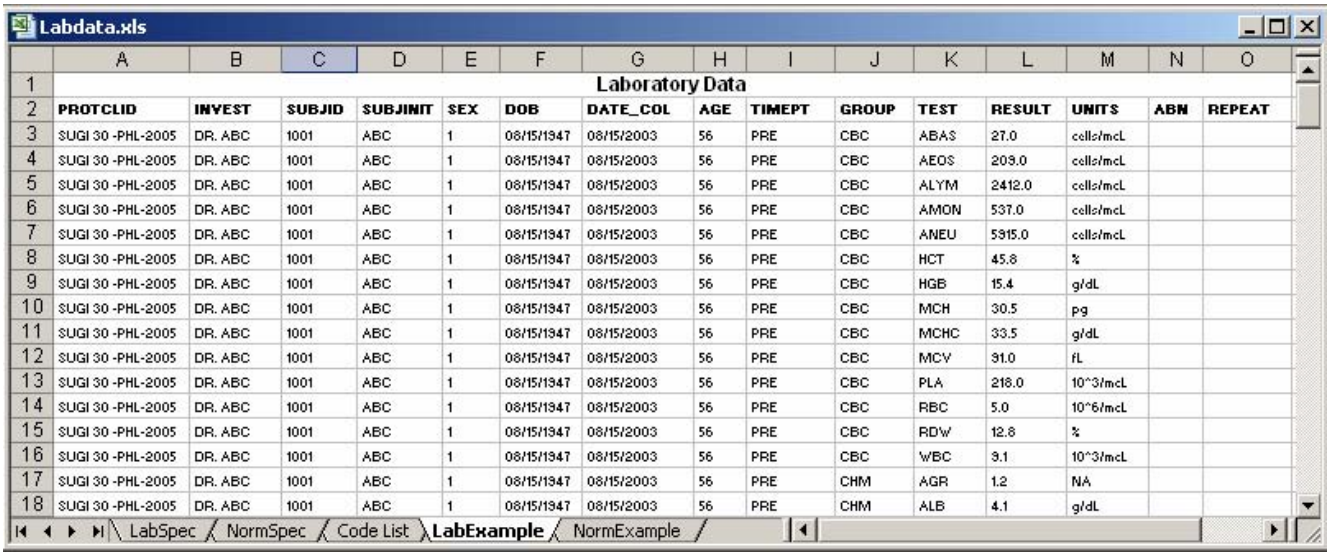


Labdata.xls

	A	B	C	D	E
1				**Code List**	
2	**Value**	**Start**	**End**	**Label**	
3	SEX	1	1	Male	
4		2	2	Female	
5		3	3	Both	
6	$TGROUP	CBC	CBC	Blood Analysis	
7		CHM	CHM	Chemical Analysis	
8		DRG	DRG	Drug Analysis	
9		URN	URN	Urine Analysis	
10	$TCODE	WBC	WBC	White Blood Cells	
11		AGR	AGR	Albumin/Globulin Ratio	
12		ALB	ALB	Albumin	
13		ALKP	ALKP	Alkaline Phosphatase	
14		BUN	BUN	Blood Urea Nitrogen	
15		CAL	CAL	Calcium	
16		CHL	CHL	Chloride	
17		CHOL	CHOL	Cholesterol	
18		CRE	CRE	Creatinine	
19		GGT	GGT	Gamma-glutamyltransferase	
20		GLOE	GLOB	Globulin	
21		GLU	GLU	Glucose	
22		PHO	PHO	Phosphate	
23		POT	POT	Potassium	
24		PRO	PRO	Total Protein	
25		SGOT	SGOT	Aspartate Aminotransferase	
26		SGPT	SGPT	Alanine Aminotransferase	
27		SOD	SOD	Sodium	
28		TBIL	TBIL	Total Bilirubin	
29		TRIG	TRIG	Triglycerides	
30		URA	URA	Uric Acid	
31		AMPH	AMPH	Amphetamines	
32		BARB	BARB	Barbiturates	
33		BENZ	BENZ	Benzodiazepines	
34		CAN	CAN	Cannabinoids	
35		COC	COC	Cocaine	
36		ETOH	ETOH	Alcohol Ethanol	
37		OPI	OPI	Opiates	
38		BILI	BILI	Bilirubin	
39		KET	KET	Ketones	
40		PH	PH	pH	
41		PROT	PROT	Protein	
42					
43					
44					
45					

LabSpec | NormSpec | **Code List** | L

Fig 3: Excel work sheet 3

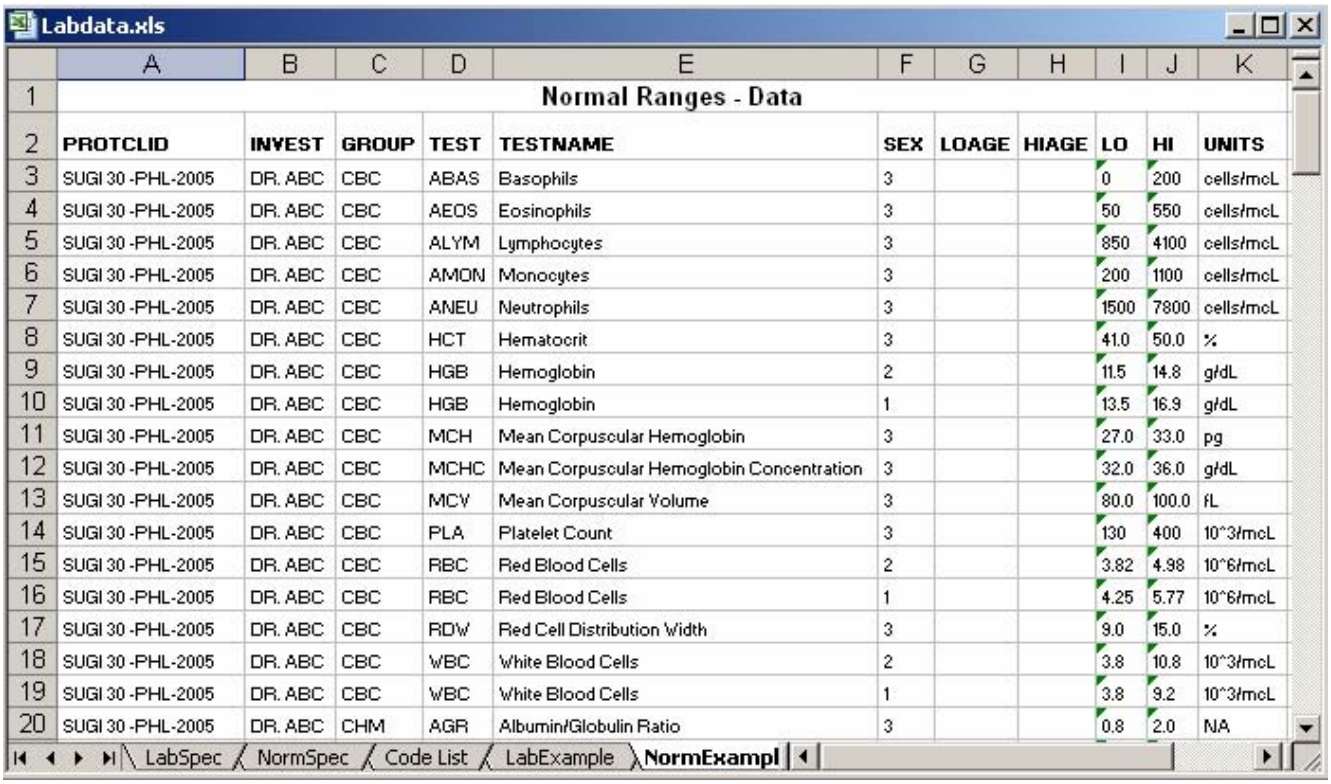


Labdata.xls

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Laboratory Data														
2	PROTCLID	INVEST	SUBJID	SUBJINIT	SEX	DOB	DATE_COL	AGE	TIMEPT	GROUP	TEST	RESULT	UNITS	ABN	REPEAT
3	SUGI 30 -PHL-2005	DR. ABC	1001	ABC	1	08/15/1947	08/15/2003	56	PRE	CBC	ABAS	27.0	cells/mcL		
4	SUGI 30 -PHL-2005	DR. ABC	1001	ABC	1	08/15/1947	08/15/2003	56	PRE	CBC	AEOS	209.0	cells/mcL		
5	SUGI 30 -PHL-2005	DR. ABC	1001	ABC	1	08/15/1947	08/15/2003	56	PRE	CBC	ALYM	2412.0	cells/mcL		
6	SUGI 30 -PHL-2005	DR. ABC	1001	ABC	1	08/15/1947	08/15/2003	56	PRE	CBC	AMON	537.0	cells/mcL		
7	SUGI 30 -PHL-2005	DR. ABC	1001	ABC	1	08/15/1947	08/15/2003	56	PRE	CBC	ANEU	5915.0	cells/mcL		
8	SUGI 30 -PHL-2005	DR. ABC	1001	ABC	1	08/15/1947	08/15/2003	56	PRE	CBC	HCT	45.8	%		
9	SUGI 30 -PHL-2005	DR. ABC	1001	ABC	1	08/15/1947	08/15/2003	56	PRE	CBC	HGB	15.4	g/dL		
10	SUGI 30 -PHL-2005	DR. ABC	1001	ABC	1	08/15/1947	08/15/2003	56	PRE	CBC	MCH	30.5	pg		
11	SUGI 30 -PHL-2005	DR. ABC	1001	ABC	1	08/15/1947	08/15/2003	56	PRE	CBC	MCHC	33.5	g/dL		
12	SUGI 30 -PHL-2005	DR. ABC	1001	ABC	1	08/15/1947	08/15/2003	56	PRE	CBC	MCV	91.0	fL		
13	SUGI 30 -PHL-2005	DR. ABC	1001	ABC	1	08/15/1947	08/15/2003	56	PRE	CBC	PLA	218.0	10^3/mcL		
14	SUGI 30 -PHL-2005	DR. ABC	1001	ABC	1	08/15/1947	08/15/2003	56	PRE	CBC	RBC	5.0	10^6/mcL		
15	SUGI 30 -PHL-2005	DR. ABC	1001	ABC	1	08/15/1947	08/15/2003	56	PRE	CBC	RDW	12.8	%		
16	SUGI 30 -PHL-2005	DR. ABC	1001	ABC	1	08/15/1947	08/15/2003	56	PRE	CBC	WBC	9.1	10^3/mcL		
17	SUGI 30 -PHL-2005	DR. ABC	1001	ABC	1	08/15/1947	08/15/2003	56	PRE	CHM	AGR	1.2	NA		
18	SUGI 30 -PHL-2005	DR. ABC	1001	ABC	1	08/15/1947	08/15/2003	56	PRE	CHM	ALB	4.1	g/dL		

LabSpec | NormSpec | Code List | **LabExample** | NormExample

Fig 4: Excel work sheet 5



Labdata.xls

	A	B	C	D	E	F	G	H	I	J	K
1	Normal Ranges - Data										
2	PROTCLID	INVEST	GROUP	TEST	TESTNAME	SEX	LOAGE	HIAGE	LO	HI	UNITS
3	SUGI 30 -PHL-2005	DR. ABC	CBC	ABAS	Basophils	3			0	200	cells/mcL
4	SUGI 30 -PHL-2005	DR. ABC	CBC	AEOS	Eosinophils	3			50	550	cells/mcL
5	SUGI 30 -PHL-2005	DR. ABC	CBC	ALYM	Lymphocytes	3			850	4100	cells/mcL
6	SUGI 30 -PHL-2005	DR. ABC	CBC	AMON	Monocytes	3			200	1100	cells/mcL
7	SUGI 30 -PHL-2005	DR. ABC	CBC	ANEU	Neutrophils	3			1500	7800	cells/mcL
8	SUGI 30 -PHL-2005	DR. ABC	CBC	HCT	Hematocrit	3			41.0	50.0	%
9	SUGI 30 -PHL-2005	DR. ABC	CBC	HGB	Hemoglobin	2			11.5	14.8	g/dL
10	SUGI 30 -PHL-2005	DR. ABC	CBC	HGB	Hemoglobin	1			13.5	16.9	g/dL
11	SUGI 30 -PHL-2005	DR. ABC	CBC	MCH	Mean Corpuscular Hemoglobin	3			27.0	33.0	pg
12	SUGI 30 -PHL-2005	DR. ABC	CBC	MCHC	Mean Corpuscular Hemoglobin Concentration	3			32.0	36.0	g/dL
13	SUGI 30 -PHL-2005	DR. ABC	CBC	MCV	Mean Corpuscular Volume	3			80.0	100.0	fL
14	SUGI 30 -PHL-2005	DR. ABC	CBC	PLA	Platelet Count	3			130	400	10^3/mcL
15	SUGI 30 -PHL-2005	DR. ABC	CBC	RBC	Red Blood Cells	2			3.82	4.98	10^6/mcL
16	SUGI 30 -PHL-2005	DR. ABC	CBC	RBC	Red Blood Cells	1			4.25	5.77	10^6/mcL
17	SUGI 30 -PHL-2005	DR. ABC	CBC	RDW	Red Cell Distribution Width	3			9.0	15.0	%
18	SUGI 30 -PHL-2005	DR. ABC	CBC	WBC	White Blood Cells	2			3.8	10.8	10^3/mcL
19	SUGI 30 -PHL-2005	DR. ABC	CBC	WBC	White Blood Cells	1			3.8	9.2	10^3/mcL
20	SUGI 30 -PHL-2005	DR. ABC	CHM	AGR	Albumin/Globulin Ratio	3			0.8	2.0	NA

LabSpec | NormSpec | Code List | LabExample | **NormExampl**

Fig 5: Excel work sheet 5