BIOS 511 Lab 12
DATA Step Programming and Using Macros in SAS

Please read the following instructions carefully before beginning this lab.

- Each task should be completed in a separate SAS program with the programs named lab-12-PID-task-X.sas where PID is your student PID number and X is the task number. Please make sure to include appropriate headers in the SAS programs.

- Task 2 requires that you write and make use of a SAS macro named CODEBOOK which *must* be stored in a SAS program named codebook-PID.sas where PID is your PID number (the only contents of this program will be the macro definition). This macro *must* be accessed in the task 2 SAS program using a %INCLUDE statement.

- You *must* use the following folder structure for your files for this lab:

    - ROOT
        - parent directory for your lab 12 files
    - ROOT\programs
        - location where task 1 and 2 programs are stored and their corresponding SAS logs
    - ROOT\output
        - location where output PDF files should be written by your SAS program
    - ROOT\macros
        - location where the codebook-PID.sas macro should be stored for access by the task 2 SAS program
    - ROOT\data
        - location where the ADSL dataset is written by the task 1 SAS program

- You will upload the SAS programs (including the SAS macro program), SAS logs for the task 1 and 2 programs, ADSL.sas7bdat, and PDF output files to document completion of the lab. IN ADDITION to uploading the individual files, you *must* upload a ZIP file of everything in the root directory (so we can verify that you used the appropriate folder structure). The ZIP file should be named lab-12-PID.ZIP.

- The submitted logs should reflect clean runs of the complete SAS program (i.e., they should not contain log messages from when the program was being developed). There will be no log for the SAS macro program.

- Helpful suggestions:

    - When first writing the codebook macro definition it is advisable to develop the macro as a part of the task 2 SAS program (i.e., leave the macro definition in the task 2 program). Then, move the macro definition into a separate file (as one would do to make it accessible to others or for use by multiple programs) as a final step.

    - Remember to use the MPRINT, MLOGIC, and SYMBOLGEN options to help debug your macro code as you develop it. These options can be turned off and turned on as needed. It is advisable to turn them off when not needed (or when the macro is complete) as these options add a substantial amount of additional information in the SAS log that is only helpful when debugging the macro.

Logs that contain ERRORs, WARNINGs, etc. will result in a point deduction of *at least 10 points*.

BIOS 511 Lab 12
DATA Step Programming and Using Macros in SAS

Task 1: Write a SAS program that creates a permanent dataset named ADSL (which stands for **S**ubject-**L**evel **A**nalysis **D**ataset). The dataset should contain only the following variables which should be ordered in the dataset from left to right as shown below:

Variables in ADSL Dataset				
#	**Variable**	**Type**	**Len**	**Label**
1	USUBJID	Char	30	Unique Subject Identifier
2	ARMCD	Char	20	Planned Arm Code
3	ARM	Char	50	Description of Planned Arm
4	AGE	Num	8	Age
5	AGECAT	Char	5	Age Category
6	COUNTRY	Char	20	Country
7	PCMAX	Num	8	Maximum Plasma Concentration
8	DIABP_CHANGE	Num	8	Change in Diastolic Blood Pressure
9	SYSBP_CHANGE	Num	8	Change in Systolic Blood Pressure
10	HR_CHANGE	Num	8	Change in Heart Rate
11	WGT_CHANGE	Num	8	Change in Weight

1. The variables USUBJID, ARMCD, ARM, AGE, and COUNTRY exist in the DM dataset.

2. The AGECAT variable should be programmed to take values “<45”, “45-55”, and “>=55” based on the value of the AGE variable.

3. The PCMAX variable should be the maximum plasma concentration value (considering all time points) for the subject based on the plasma concentration data in the PC dataset. This dataset was distributed with Lab #10 (which was optional). See Lab #10 for more details on this dataset and to download the dataset if you have not already. *Note: Impute concentration values lower than the limit of quantification to the lower limit of quantification prior to determining the maximum concentration value for a subject.*

4. The DIABP_CHANGE, SYSBP_CHANGE, HR_CHANGE, and WGT_CHANGE variables should all be programmed in a similar way. These values should be programmed as Y - X where Y and X are defined as follows:

    a. Y = Average value of all measures that occurred *after* the Week 0 visit.

    b. X = Average value of all measures that occurred *on or before* the Week 0 visit.

    Thus, the subject-specific values of Y and X will be an average of one or more values for that subject.

BIOS 511 Lab 12
DATA Step Programming and Using Macros in SAS

Task 2: Write a SAS program that creates two codebooks that closely resemble the files ADSL_CODEBOOK.PDF and DM_CODEBOOK.PDF which have been distributed along with the lab. The task 2 program should resemble the following when complete (written for SDM):

```
ods html close;
%let root        = P:\Teaching\BIOS-511-FALL-2018\2018-11-20-lab-12-solution;
%let echoDat     = C:\Users\psioda\Documents\GitHub\BIOS-511-FALL-2018\data\echo;
%let analysisDat = &root.\data;
%let outputPath  = &root.\output;
%let macroPath   = &root.\macros;

libname echo "&echoDat."      access=read;
libname out  "&analysisDat." access=read;

ods noptitle; option nonumber nodate;

%include "&macroPath.\codebook.sas";

ods pdf file="&outputPath.\ADSL CODEBOOK.pdf" style=sasweb;
  %codebook(lib=out,ds=adsl,maxVal=15);
ods pdf close;

ods pdf file="&outputPath.\DM_CODEBOOK.pdf" style=sasweb;
  %codebook(lib=echo,ds=dm);
ods pdf close;
```

- The CODEBOOK macro should be included in the SAS program using the %INCLUDE statement.

- The CODEBOOK macro should have three non-positional parameters:

  - LIB = SAS library that contains the dataset for which the codebook is to be created

  - DS = SAS dataset for which the codebook is to be created

  - MAXVAL = Maximum number of unique values for a character variable such that all values are printed. If the number of unique values is more than MAXVAL, then only the most frequent &MAXVAL values should be printed. The default value of MAXVAL should be set to 10 (and thus the default value will be used for the DM codebook.

- Much of the sas code required for the CODEBOOK macro can be taken from the SAS program developed in lecture during recent class meetings. The program can be found in the class GITHUB repository (BIOS-511-FALL-2018/programs/2018-11-05-lecture-18/04-Understanding-Call-Symput.sas). Some modifications will be required.

  - The codebook should be constructed based on the order of the variables (i.e. columns) when viewed/read from left to right in the SAS dataset (i.e., not based on alphabetical order).

  - For character variables, you will need to modify the code from class to determine how many unique values exist for the variable being processed. If that number exceeds &MAXVAL, only a subset of the frequency table should be printed (i.e., the &MAXVAL most frequent values). This is a necessary modification to prevent character variables such as USUBJID from contributing an unnecessarily long frequency table.

    Note: To arrange the frequency table in descending order of frequency of occurrence use the ORDER= option for the PROC FREQ statement.

- The solution codebooks have modified the PDF bookmarks so that the variable name/label is a part of the bookmark. Try to do this if you want an extra challenge!