# BIOS 511 Lab 13
# DATA Step Programming + SQL + Macros

**Please read the following instructions carefully before beginning this lab.**

- This lab should be completed in a single SAS program named lab-13-PID.sas where PID is your student PID number. Please make sure to include an appropriate header in the SAS program.

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

    - ROOT
        - parent directory for your lab 13 files
    - ROOT\programs
        - location SAS program is stored with corresponding SAS log
    - ROOT\output
        - location where output PDF files should be written by your SAS program
    - ROOT\data
        - location where the QC_DATES.CSV should be saved

- You will upload the SAS program, SAS log, and PDF output files to document completion of the lab. Upload the pdf files as a ZIP file. DO NOT UPLOAD THE INDIVIDUAL FILES AS THERE WILL BE 30. The ZIP file should be named lab-13-PID.ZIP.

- The submitted logs should reflect a clean run of the complete SAS program (i.e., it should not contain log messages from when the program was being developed).

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

# BIOS 511 Lab 13
# DATA Step Programming + SQL + Macros

**Task 1:** Vital sign data for the ECHO trial is extracted from electronic medical records and then recorded onto a paper form that is mailed to a central location for entry into the ECHO trial database. Using the paper form as the source document, the data are entered into the ECHO trial database manually. Though trained personnel complete these tasks, there are always mistakes!

To verify the quality of the vital sign data for the ECHO trial, study staff will assess the accuracy of the entered data against the source electronic medical records during site inspections. The study statistician has randomly selected two periods of time for each enrolling site and written the start and end dates for each time period into a file named QC_DATES.CSV. An excerpt from the beginning of the QC_DATES.CSV file is shown below:

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | ECHO Trial Quality Control Period | | | | |
| 2 | | Site | Start | End | QC |
| 3 | Country | Number | Date | Date | Period |
| 4 | CAN | 31 | 27-Jan-15 | 27-Apr-15 | 1 |
| 5 | CAN | 31 | 21-May-15 | 19-Aug-15 | 2 |
| 6 | CAN | 32 | 17-Jan-15 | 17-Apr-15 | 1 |
| 7 | CAN | 32 | 9-May-15 | 7-Aug-15 | 2 |
| 8 | CAN | 33 | 2-Mar-15 | 31-May-15 | 1 |
| 9 | CAN | 33 | 24-Jun-15 | 22-Sep-15 | 2 |

The first period for site 31 (from Canada) is from the 27th of January 2015 to the 27th of April 2015 (inclusive). The second period is from the 21st of May 2015 to the 19th of August 2015.

As the study programmer, you are tasked with producing *a separate quality control (QC) report for each site* that includes the entered vital sign data for all subjects enrolled at the site in either of the two study periods. A subject is considered to have enrolled at a site during a given period if the subject's informed consent date (DM.RFICDTC) is between the start and end date for that period (inclusive).

The QC reports that you create should be named XXX_YYY_VITAL_SIGNS.pdf where XXX is USA, CAN, or MEX and YYY is the three-digit zero-padded site number (i.e. 020 not 20). The structure of the QC reports is illustrated below for USA site 020. Using this report, study staff will verify the error rate of data extraction/entry is acceptable.

**Site 020 Vital Sign Data for Select Subjects**

**Subject Number=005**

| Visit Name | Diastolic Blood Pressure | Height | Heart Rate | Systolic Blood Pressure | Weight |
|---|---|---|---|---|---|
| Screening | 97 | 163.1 | 57 | 134 | 55.2 |
| Week 0 | 101 | . | 55 | 134 | 47.7 |
| Week 8 | 89 | . | 63 | 132 | 46.5 |
| Week 16 | 94 | . | 56 | 137 | 52.1 |
| Week 24 | 98 | . | 58 | 132 | 26.0 |
| Week 32 | 97 | . | 62 | 132 | 28.7 |

**Subject Number=006**

| Visit Name | Diastolic Blood Pressure | Height | Heart Rate | Systolic Blood Pressure | Weight |
|---|---|---|---|---|---|
| Screening | 96 | 163.4 | 57 | 131 | 51.5 |
| Week 0 | 94 | . | 63 | 133 | 60.3 |

# BIOS 511 Lab 13
# DATA Step Programming + SQL + Macros

High-Level Step-by-Step Programming Guide:

[1] Either using PROC IMPORT or a DATA step, read the contents of the QC_DATES.CSV file into a SAS dataset named WORK.RANGES. This dataset will have one observation per ECHO trial site and QC period. The DATEX (for integer X) *informat* will be needed for reading in the dates as numeric variables. Be sure that the date variables are properly read into a numeric format and visually compare WORK.RANGES to QC_DATES.CSV before proceeding.

Hint: To read in the date text in the CSV file as numeric data, you can use an in-line *informat* or an INFORMAT statement. If you use an INFORMAT statement, your DATA step would look like the following:

```
data ranges;
 infile "<path to CSV file>" <options>;
 informat X Y informatA. …;
 input <list of variables in column order possibly using $ after some>;
 format X Y formatA.;
run;
```

Note that the INFORMAT statement works like the using an *informat* with the INPUT function. For each variable listed in the INFORMAT statement one would list an *informat* (followed by a period) that provides an instruction for how to read the non-numeric data from the CSV file into a numeric SAS variable. Not all variables need to be read in with informats, only those that are non-numeric in the CSV file and for which the SAS data will be numeric (i.e., dates in the case).

Using in-line *informats* simply condenses the notation by combining the INFORMAT and INPUT statements. One can do the following:

```
data ranges;
 infile "<path to CSV file>" <options>;
 input … X:informatA. Y:informatA. …;
 format variable2 variable3 formatA.;
run;
```

[2] Using a DATA step (that need not create a new dataset), create a macro variable named NUMSITES that stores the number of sites in the dataset WORK.RANGES and a set of macro variables SITE1-SITEX (for integer X representing the number of distinct sites) that stores the site identification numbers for the study sites included in the dataset WORK.RANGES. If this is done correctly &NUMSITES should resolve to 30 (for this QC_DATES.CSV file) and &SITE1 should resolve to a site ID such as 31 or 031 (depending on how you write your program).

[3] Write a PROC SQL step that performs an INNER JOIN of the ECHO.DM and WORK.RANGES datasets that only keeps observations where the following are all true: (1) country value in ECHO.DM matches WORK.RANGES, (2) site ID (extracted from USUBJID) in ECHO.DM matches site ID from WORK.RANGES, and (3) informed consent date from ECHO.DM falls between the QC period start and end dates (inclusive). This PROC SQL step should create a new dataset named SUBJECTS and must keep at minimum the country, site, and subject number of all subjects enrolled during a QC period (these data will be needed for the report).

[4] Write a macro named GEN_REPORT that does the following:

> [i] Performs a %DO loop over the macro variable index J ranging from 1 to &NUMSITES.
>
> [ii] Within each iteration of the %DO loop, do the following:
>
> > [a] Create a temporary dataset that contains observations for the subjects selected for QC from the site identified by &&SITE&J. Call this dataset WORK.TEMP.

[b] Create macro variables for the country and formatted site number (they are needed for title statements). These macro variables can be called anything (e.g., CNT and SITE) and can be overwritten with each new loop iteration.

[c] Read in and merge the ECHO.VS dataset with the WORK.TEMP dataset keeping the vital sign data (e.g. height, weight, heart rate, diastolic/systolic blood pressure) for only those subjects selected for QC at the site. You can create a new dataset named WORK.TEMP2 or simply overwrite WORK.TEMP (the former strategy may be more useful for debugging). You can complete this step with a DATA step merge or a SQL join.

[d] Using a DATA step or PROC TRANSPOSE, transform the dataset from [c] to have one observation per subject and visit (i.e., each vital sign should become a column). This new dataset can overwrite WORK.TEMP or create a third temporary dataset.

[e] Using a PROC PRINT step wrapped in ODS code to generate a PDF file, print the data for each report into a PDF file named using the macro variables from [b]. Note, to match the solution example the PROC PRINT step will need to use a BY statement. Recall these PDF files should be named XXX_YYY_VITAL_SIGNS.pdf where XXX is USA, CAN, or MEX and YYY is the three-digit zero-padded site number (i.e. 020 not 20). Check one report against those provided with the lab to verify your report is correct.

- Helpful Notes:
  - Since the GEN_REPORT macro is being used to repeatedly generate what you expect to be one-time reports, there is no need to store the macro outside of the program that generates the reports.
  - When first writing the GEN_REPORT macro, you can initially ignore the macro aspect altogether and simply assign the %DO loop index variable J a value with a %LET statement. For example,

    Instead of this:

```
%macro gen_report;
      %do i = 1 %to &numsites.;
            .
            .
            .
      %end;
%mend;
%gen_report;
```

    Start with this:

```
/*%macro gen_report;*/
/*    %do j = 1 %to &numsites.;*/
 %let j = 1;
            .
            .
            .
/*    %end;*/
/*%mend;*/
/*%gen_report;*/
```

If you take this approach during program development, you can run the code as you go (i.e., run the . . . part) to verify that (for the chosen site) the code is functioning correctly. Once the code for report generation is complete, you can comment out the %LET statement and uncomment the macro code to run the reports for all sites.

- 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.