Programming HW #3

Specifications:

- You are allowed at most three `DATA` steps and thirteen `PROC` steps to complete this assignment.
- The data you need to read in is in the `Data\BookData\ClinicalTrialCaseStudy` folder - associate a *fileref* named RawData with this location. The data sets I've provided you for validation purposes are in the `Results` folder - associate a *libref* named Results with this location. As usual, your *libref* for storing your results must be named HW3.
- Read in the raw files associated with the 3 Month visits for Sites 1, 2, and 3. (E.g., the first file you'll need is Site 1, 3 Month Visit.txt.) Variable order is the same in all files, but you need to determine how to read them in.
- I've provided my three SAS data sets, e.g. HW3DugginsSite1, and their descriptor portions, e.g. HW3DugginsPosition1, for you to reference throughout the assignment. The variables in the raw files are in the same order as the variables in my data sets. For Site 3, which is fixed-field data, the variables start in the following columns. There is no padding between variables; e.g., variable 2 starts in column 8 and ends in column 58 since variable 3 begins in column 59.

Variable #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Start Column	1	8	59	62	63	73	83	86	89	95	98	108	119	124	125	128	130

- As usual, your goal is to reproduce one or more pieces of commonly requested output using techniques acquired in class. Now that we know how to electronically validate results, this and future assignments require some or all of your work to be validated. As in the past, I've put the particular components that are not obvious from the report below.
  - When reading the data for the second site, write the IB to the log for every record.
  - Site 3 has had some issues recording the value of Pulse for patients in the past. Write the values of Subject and Pulse from the PDV to the log for every record. Make sure to include the name of the variable, e.g., Pulse = 75. (This allows a programmer to review the log to ensure everything is fine without having to look at any output.)
  - After reading in the data, electronically validate your results against mine. We aren't quite ready to validate the descriptor portions electronically yet, so start by manually comparing those since, if those do not match, there is no need to worry about the data portions. Once your descriptor portions match, validate the data portions.
  - I repeat – don't bother validating the data sets if they have different descriptor portions! We only compare variables of the same type with the same name, so it is important your **attributes** match before your **contents** undergo validation!
  - For all electronic validation, use the process described in class: create data sets that contain any differences and work until those data sets are empty; Do not produce any printed output; Use an ABSOLUTE method with a CRITERION of 1E-10 for numeric differences.
  - You are creating three files - PDF, RTF, and PowerPoint. For the latter one, the ODS destination name is simply POWERPOINT and the file extension is PPTX. The PDF uses the PRINTER style, RTF uses the SAPPHIRE style, and the PowerPoint uses the POWERPOINTDARK style.
  - Name your files the way we did before, including your name as part of the file. For example, Tony Stark would create a file called HW3 Stark 3 Month Clinical Report.
  - All footnotes are 10pt font and left justified.
  - The three reports are not identical - only analysis results are sent to the PowerPoint file. (You already know how to select output for specific destinations based on HW2, so this is more practice of that skill.) The RTF and PDF files include the same content, but you'll notice the files do not look the same.
  - At several points in the report, I have applied a custom format to the Blood Pressure variables (DBP and SBP). This format is not provided, so you will need to create it for yourself based on the information in the report. *To combine the skills from HW1 and HW2, HW3 requires you to save the format in your HW3 library and apply it from there.*
  - Certain options for a destination, such as setting COLUMNS=2 in a PDF, can be changed as needed by adding additional ODS PDF statements with the required options. (Just don't issue the FILE= option again!)
  - Now that you have seen and used macro variables, we'll use them to cut down on repetitious code and to include variable content in a literal token
    - Each data set you create needs the same attributes, so use a macro variable named VarAttrs to store all the coding elements for your ATTRIB statement. Use the macro variable any time you need to apply the attributes.

- Since the same visit is needed from all sites and referenced in many headers and footers, use a macro variable named Visit to both (a) select the correct visit file from the Raw library and (b) insert the correct visit name into those headers and footers. (Note this is *Visit* and not *Site*. So, you are looking to sub in values like Baseline, 3 Month, 6 Month, etc. If you've done this correctly, a new set of reports for a different visit can be generated by changing this one line of code and rerunning your program. If I were you, I would test it by doing just that!)
- The sorting uses the same hierarchy and order, so use a macro variable called ValSort to control sort details.
- Since we always use the same settings for our PROC COMPARE, use a macro variable named CompOpts to apply all the settings that are common across the COMPARE steps.
- Use the appropriate automatic system macro variables to insert your user ID and time stamp into the footnotes for the output. (E.g., that's why mine says Jonathan with the last date that output file was produced.)

Note: I know the PowerPoint file looks terrible! This is because (a) we are depending on default font sizes for titles and you can see PDF, RTF, and PowerPoint have different defaults and (b) we're depending on the default PROC MEANS and PROC FREQ templates which are not optimized for a PowerPoint presentation.

In practice, delivering directly to PowerPoint is typically only useful for either (a) graphics or (b) simple tables. For more customization, we would typically either (a) deliver to PowerPoint on its own so that we can use specific options or, in more detailed cases, (b) use ODS tools to produce highly-customized outputs. I just wanted y'all to see the differences that we might need to control in different destinations.