**In-Class Exercise #2**

**Basic SAS**

**Tasks**

1. Data Import/Export
2. Handling variable TYPEs (CHARACTER and NUMERIC)
3. Become familiar with handling DATE variables
   1. Convert Character or Numeric variables to SAS DATE variables
   2. SAS DATE Formats
   3. SAS DATE Functions
4. Know how to get basic statistics
5. Learn how to run a SAS program on the WRDS.

We will run following programs as part of this in-class exercise:

* sas\_unix.sas (set up to run on the WRDS server)

Try to run the program on WRDS before class. To run sas\_unix.sas (i.e., the WRDS program), put the two input files in the same folder as the program. Then, type in the following command:

>>qsas sas\_unix.sas

If the program runs without errors, the log and lst files will be found in the same directory as the SAS program. If there are errors, you will typically only see the log file.

**Class notes on basic SAS programming**

1. Mapping to a Permanent folder
   1. Libname junyoup ‘/home/kaist/tigerz7/fin514’;
   2. SAS also provides a "Work" folder by default;
   3. Any dataset saved in the work folder is deleted when SAS is closed
2. Two important things to remember in writing a SAS program
   * 1. Put a semi-colon(”;”) at the end of each SAS statement
     2. Comments are created with a “\*” followed by a semi-colon or “/\* \*/”
3. SAS files
   * 1. sas\_unix.sas (your SAS program)
     2. sas\_unix.log (contains your SAS program with messages from SAS – including error messages)
     3. sas\_unix.lst (the SAS output file with the results of your program)
4. Main parts of a SAS program
   1. DATA step (define / create variables)
   2. PROC step (SAS procedures)
5. DATA step
   * 1. DATA statements are used to create or modify a SAS dataset (e.g., “dataset1” in the above example)
     2. Observations are read into the SAS dataset either internally (e.g., with the CARDS statement) or from an external file (non-SAS: .csv, .txt …) with the INFILE statement
     3. For data with more than 132 columns, include the LRECL statement (e.g., LRECL=1000) at the end of the INFILE statement
     4. The INPUT statement give the names of the variables and defines as character or a number
6. SAS datasets
7. SAS datasets have rows and columns (like a spreadsheet)
8. Each column represents another variable (VAR)
9. Each row is an observation (OBS)
10. Missing values in SAS are denoted with a “.” (a dot) if a number variable or “ ” (a blank) if character
11. The above examples have four variables (columns) and four observations (rows)
12. Input data in fixed versus list (free) format
    * 1. Input data in fixed format (i.e., aligned in straight columns) is easiest to work with. Notice that the input statement tells the program where to find the values for the different variables
      2. List (or free) format input is used for data that is not in straight columns. For instance, **space delimited** (the default in SAS, or DLM = “ ”) works on the data for dataset2
      3. Comma delimited is another common format for input files (include DLM = “,” in the INFILE statement) and is the data for dataset4
13. Notes on variables
    * + 1. Variables are either a character or a number
        2. SAS does not distinguish between real and integer
        3. By default, all variables are numbers
        4. Character variables are distinguished by a “$” in the INPUT statement following the variable name
        5. A number (like 123) can be a character or a number (e.g: ID variable in dataset1)
        6. Anything with a letter or special character (e.g. #) has to be character
        7. Math operations can be performed only on number variables
        8. Math operations and the creation of new character and number variables can only be performed in the DATA step of the program
14. PROC step
    * + - 1. The PROC step is the main advantage of using SAS – it allows the user to do simple things (e.g., print observations) or more complex things (e.g., run regressions)
          2. PROC statements are followed with RUN statements
          3. All PROC statements use the current active SAS dataset unless specifically designated (e.g., PROC PRINT DATA=dataset1; RUN;). It is good form to include a DATA statement in each PROC statement.
          4. We’ll initially spend most of our time with the DATA step
15. Running your SAS program on the WRDS system

Reference (한글 링크들)

* 1. Infile & cards statement:

https://communities.sas.com/t5/SAS-Tech-Tip/SAS-%ED%99%9C%EC%9A%A9-%EB%85%B8%ED%95%98%EC%9A%B0-INPUT-%EB%AC%B8/ta-p/791789

* 1. Inflie 옵션들 (missover, lrecl, firstobs and etc): https://www.sasbigdata.com/entry/SAS-sas-infile%EC%97%90-%EB%8C%80%ED%95%B4%EC%84%9C-%EC%95%8C%EC%95%84%EB%B3%B4%EA%B8%B0%EC%98%B5%EC%85%98-%EB%93%B1