Introduction to SAS

Overview

SAS Window Environment.
Explorer window. Access libraries (and the files in your computer).
Editor window. Write and edit programs. You can have more than one editor window.
Log window. Cumulatively record the submitted program statements, including any resulting warning.
Results viewer window. Displays the previous results (in html by default).
Results window. Cumulatively saves the results of submitted statements.

Tips.

- The toolbar is context sensitive. If you don't see the button you'd like, make sure the corresponding window is active.

SAS programs are a sequence of statements executed in order. A statement gives instructions to the software and must be appropriately placed. An analogy to a SAS program is a trip to the bank.

```
I would like to make a withdrawal ;
My account number is 0937 ;
I would like $200 ;
Give me five 20s and two 50s ;

(Example taken from "The Little SAS Book")
```

Basic rules.

- Every SAS statement ends with a semicolon.
- Statements can continue on the next line.
- Statements can be at the same line.
- No distinction between upper and lower case.

Tips.

- Build complicated programms one bit at a time.
- One missing letter is enough to geneate errors (and that's fine: the computer need precise instructions).
- Save your code (often!).

DATA and PROC commands.

The two basic commands used in SAS are DATA (which creates a data set) and PROC (which specifies a procedure to be performed).

```
* A simple example taken from "The Little SAS Book";
DATA distance;             /* create a data set called "distance" */
Miles = 3;                 /* within it, create a variable miles with value 3 */
Kilometers = 1.61 * Miles; /* and a variable kilometers, that is a linear transformation of miles */
RUN;

* Show the data set ;
PROC PRINT DATA = distance; RUN;
```

Basic rule.

- A step ends when SAS encounters a new step (i.e. another DATA or PROC statement), or a RUN statement.

Tips.

- The shortcut F3 will run any selected lines.

Adding comments.

- Comments are ignored by the software. They are meant to help you document the code.
- General comments start with an asterisk (*) and end with a semicolon (;)
- Inside statements, comments can be enclosed by /* and */

Data

SAS data sets. Data sets (or tables) are composed of observations (or rows) and variables (or columns).

Data types. Base SAS has only two types: numeric and character.
Missing data. Missing character data are represented by blanks, missing numeric data are represented by a single period (.).
Size. The maximum number of observations and variables is limited only by the computer's storage capacity.
Naming conventions. Variable names should have less than 32 characters, be composed only of letters and numerals and start with letters or underscore (_).
Descriptor. Data sets metadata include its name, the creation date, and the SAS version. Variable metadata includes its name, label (if any), type (numeric or character), length (or storage size), and position within the data set.

A SAS data library is a collection of data sets (which are called members of the library) which are stored in a same location.

- All SAS data sets have a two-level name such as WORK.BANANA.
- The first level is called its libref (short for library reference), which is usually associated to a directory or folder in your computer.
- And the second level is the member name, which uniquely identifies the data set within the library.
- Libref names cannot be longer than 8 characters, member names can be up to 32 characters.
- Temporary data sets are stored into the built-in WORK library, which is erased when you exit SAS.

LIBNAME defines the name of the library reference and maps it to an specific folder.

```
LIBNAME my_library 'C:\MySASLib'; RUN;
```

Importing and manipulating data

PROC IMPORT reads excel files (and other file types) into SAS data sets.

```
* Set libref ;
LIBNAME my_library 'C:\MySASLib'; RUN

* Read excel spreadsheet ;
PROC IMPORT
DATAFILE = 'C:\path\datafile.xls'
OUT = my_library.datafile
DBMS = XLS; /* could also be XLSX, CSV or TAB */
RUN;
```

DATA can create new data sets by modifying or subsetting other existing ones using logical operators.

```
* Create a new data set which includes new variables ;
DATA my_library.datafile_new;
SET my_library.datafile_original;
variable_A = 0;
variable_B = log(variable_X);
variable_C = variable_X + variable_B;
RUN;

* Create a new data set which includes new variables using complex rules ;
DATA my_library.datafile_new;
SET my_library.datafile_original;
variable_A = 0;
IF variable_X = 1 then variable_A = 1;
ELSE IF variable_Y > 1 then variable_A = 2;
RUN;

* Create a subset from the original data set ;
DATA my_library.datafile_new;
SET my_library.datafile_original;
IF variable_X > 10;
RUN;
```

Caution: if you use the same file as DATA and as SET you will overwrite that file. If the code is not correct, you can erase it.

PROC CONTENTS lists the contents of a data set.

```
PROC CONTENTS
DATA = my_library.datafile_new;
RUN;
```

PROC PRINT shows the observations of a data set.

```
* Using a specific reference ;
PROC PRINT
DATA = my_library.datafile_new;
RUN;

* Otherwise SAS will print the last file created ;
PROC PRINT;
RUN;
```

ANALYSIS

Summary statistics

PROC MEANS calculate mean, std. dev., min and max. Can be used with options **VAR** and **BY**.

```
PROC MEANS
DATA = my_library.datafile;
VAR wages;
BY educ;
RUN;
```

PROC UNIVARIATE offers a more complete picture of the variable, with more measures of center, spread and position. Can be used with options **VAR** and **BY**, as well as **HISTOGRAM / NORMAL**.

```
PROC UNIVARIATE
DATA = my_library.datafile;
VAR wages;
HISTOGRAM / NORMAL;
RUN;
```

Plots

PROC GPLOT will produce plots. Use **PLOT** for a scatterplot of y on x . Can be extended with **BY**.

```
PROC GPLOT
DATA = my_library.datafile;
PLOT wage * educ_years;
BY gender;
RUN;
```

Linear regressions

PROC REG performs a linear regression. Can include tests on the parameters.

```
PROC REG
DATA =  my_library.datafile;
MODEL y = a b c;
RUN;

TEST a = 0; RUN;
TEST a = b; RUN;
```

Sources

- Delwiche, Lora D., and Susan J. Slaughter. 2019. *The Little SAS Book: A Primer*. 6th ed. Cary, NC: SAS Institute Inc.
- UCLA - SAS Class Notes
- UCLA - REGRESSION WITH SAS - CHAPTER 1 - SIMPLE AND MULTIPLE REGRESSION