# What is SAS?

INTRODUCTION (part 2): Date 17/09/2020

# Basic steps in SAS programs

- SAS program consists of two necessary steps
  - DATA (or data) step
  - PROC (or proc) step (short for procedure)
- DATA step:
  - Typically used to read or write the data, manipulate the data, and perform calculations.
  - Begins with DATA (or data) statement.
- PROC step:
  - Typically used to process SAS datasets in analyzing.
  - Run one or more of its procedures to produce reports, summarize the data, and much more.
  - Begins with PROC (or proc) statement.
  - Display the data.
- The end of the DATA or PROC steps are indicated by:
  - RUN statement – most steps
  - QUIT statement – some steps(ex: SQL quarries)

# Basic steps in SAS programs contd…

- Example

Example1.sas

```
/* Lecture 1, example 1: Read the dataset and print */

data studgrade; /* data step: assign a name for the dataset and enter the data */
input StudID Midterm Final Grade $; /*Variables in the dataset and enter the data*/
datalines;
101 98 86 A
102 49 60 C
103 98 80 A
104 90 98 A+
105 60 80 B+
106 .  80 C-
;
run; /*end the data step*/
proc print data = studgrade; /* proc step: perform any analysis and  print the dataset.
                                Here we are only printing the dataset*/
run;
```

# Syntax and errors

- Syntax rules for SAS statements
  - Free-format – can use upper or lower case
  - Usually begin with an identifying keyword
  - Can span multiple lines
  - Always end with semicolon
  - Multiple statements can be on the same line

- Errors
  - Misspelled keywords
  - Missing or invalid punctuation (missing semi-colon)
  - Invalid options
  - Indicated in the Log window

- SAS names
  - All SAS variable names and dataset names can be no longer than 32 characters and must begin with a letter or the underscore ( _ ) character.
  - The remaining characters in the name may be letters, digits, or the underscore character

# Syntax and errors contd…

- SAS names
  - Some valid and invalid names

| Valid SAS Names |
| --- |
| Parts |
| LastName |
| First_Name |
| Ques5 |
| Cost_per_Pound |
| DATE |
| time |
| X12Y34Z56 |

| Invalid SAS Names | |
| --- | --- |
| 8_is_enough | Begins with a number |
| Price per Pound | Contains blanks |
| Month-total | Contains an invalid character ( - ) |
| Num% | Contains an invalid character (%) |

# SAS Data Sets and SAS Variables

- SAS reads data from almost anywhere (for example raw data, spreadsheets, databases).
- SAS stores the data in its own special form called a *SAS data set* (after running `data` step).
- Only SAS can read and write SAS data sets
- You don't have to worry about how SAS is storing its data or the structure of a SAS data set.
- However, it is important to understand that SAS data sets contains two parts:
  - *descriptor* portion
  - *data* portion
- *Descriptor* portion: contains the vital statistics of the data set, such as the name of the data set, the date and time that the data set was created, the number of observations and the number of variables (`proc contents`).
- *data* portion: a collection of data values that are arranged in a rectangular table (`proc print`).
- Ex: descriptor portion and the data portion of 'Studgrade' dataset.

# SAS Variables

- SAS has only two types of variables: *character* and *numeric*.
  - This makes it much simpler to use and understand than some other programs that have many more data types (for example, integer, long integer, and logical).
  - SAS determines a fixed storage length for every variable.
  - No need to think about storage lengths for numerical values most of the time.
  - Each character value (data stored as letters, special characters, and numerals) is assigned a fixed storage length explicitly by program statements or by various rules that SAS has about the length of character values.