

Efficiency of SAS Programs

General Concepts:

- 1) Avoid unnecessary program statements
- 2) Minimize Memory (Storage) requirements
- 3) Minimize INPUT/OUTPUT (I/O) operations

Specific Strategies

- I) Avoid unnecessary DATA steps

- II) Do not read files (ASCII and SAS) repeatedly, at least not more than necessary.

III) If possible, perform programming statements in the same data step that contains the INPUT stmt.

IV) Minimize INPUT operations and processing of program statements by using Conditional Input:

IV) When possible, use a WHERE stmt in a PROCEDURE, rather than create a subset of an existing data set.

V) Do not INPUT unnecessary variables.
Drop unnecessary/temporary variables.

VI) The DROP= or KEEP= dataset option is more efficient than the DROP stmt.

VII) Optimiz memory requirements by using a Length stmt.

Recall: ① Default Length of character variables is 8 Bytes.

If a variable does not require 8 Bytes to store its values use Length stmt to specify exactly how many Bytes needed

If character variable requires more than 8 Bytes to store its values, do not allocate more Bytes than necessary.

- ② Default length of numeric variable is 8 Bytes (or 64 bits). Values stored in binary (actually "two's complement"). Known as "double precision" in languages like C++, Fortran, Pascal, BASIC, PL/I.
- Values stored in exponential notation:
- sign
 - mantissa
 - exponent

IF SAS reads a variable's value using column INPUT or pointer-format INPUT (e.g. @5 ID \$3.) the length of each variable is automatically determined by SAS, and a Length stmt is unnecessary for the variable.

IF using Length stmt, place it BEFORE the INPUT stmt.

Cannot change the length of a variable that already exists in the data set.

Suggestion: Do NOT reduce the length of numeric variables. May affect result of statistical functions and Procedures.

Use character variable to store
0/1 values

VIII) Use Nested conditional structure
rather than list of If stmts

IX)

More Tutorials at www.LittleDumbDoctor.Com

X) Use Multiple OR operators instead of IN operator

XI) Create a `-NULL-` data set if the data set (or subset) will not be needed in a subsequent step.

A `-NULL-` data step executes all the program statements, but does not create a SAS data set.

XII) Save permanent SAS data sets rather than reprocessing the data, if the data set will be used at a later time.

XIII

XIV

XV)

XVI)

XVI) Use RETAIN stmt to initialize constants. (???)

XVII) Avoid unnecessary Sorts
Do multi-level Sorts

XVIII) Avoid unnecessary Sorts

Use CLASS stmt if possible

IF data is unsorted, more efficient to use CLASS stmt to obtain subset analysis.

IF data is already sorted, more efficient to use BY stmt to obtain subset analysis.

XIX) Sort only the data that needs to be sorted

Use a WHERE stmt in PROC SORT

Use a DROP = (or KEEP =)

dataset option.

~~XX~~) Use NOEQUALS option in
PROC SORT

if you do not need to keep the
original order of records in each
BY group.