

10 Ways to Optimize Your SAS Code

10 Quick Tips in 30 Minutes

Melodie Rush

Global Customer Success Principal Data Scientist

Connect with me

LinkedIn: <https://www.linkedin.com/in/melodierush>





Melodie Rush

Principal Data Scientist, SAS

Melodie works on the Customer Success Technical Team at SAS. Since joining SAS, she has developed presentations and methodology for doing many types of analysis, including data mining, machine learning, forecasting, data exploration and visualization, quality control and marketing. She has spent more than 20 years helping companies identify and solve problems in each of these analytical areas.

1.

Test your programs with the **OBS=** & **FIRSTOBS=** options

```
data complicated_program;  
  set sample_data(obs=50);  
  many, many, many more statements here;  
run;
```

This technique may not adequately test all conditions but will confirm the correctness of the overall program logic – and save time and computer resources!

FIRSTOBS= option tells SAS which observation to start with

2.

Use IF-THEN-ELSE instead of IF-IF-IF

Do this:

```
data new;  
  set old;  
  if condition then  
    some action;  
  else if condition then  
    some other action;  
  else if condition then  
    some other action;  
run;
```

Not this:

```
data new;  
  set old;  
  if condition then  
    some action;  
  if condition then  
    some other action;  
  if condition then  
    some other action;  
run;
```

Please note that this general recommendation relates to conditions that are mutually exclusive. If the conditions are not mutually exclusive, then further consideration is in order to see whether IF THEN ELSE is appropriate.

2.

Order IF THEN conditions in descending order of probability

```
data new;  
  set old;  
  if condition occurring most often then  
    some action;  
  else if condition then  
    some other action;  
  else if condition then  
    some other action;  
run;
```

3.

Minimize the number of times you read your data

Do this

```
data a b c;  
  set old;  
  if condition then  
    output a;  
  else if condition then  
    output b;  
  else if condition then  
    output c;  
run;
```

Not this:

```
data a;  
  set old;  
  [more code]  
run;  
data b;  
  set old;  
  [more code]  
run;  
data c;  
  set old;  
  [more code]  
run;
```

3.

Minimize the number of times you read your data

```
proc freq data = sashelp.shoes;  
table region / list out=region_freq1;  
table region*product / list out=region_freq2;  
table region*stores / list out=region_freq3;  
run;
```

```
proc freq data = sashelp.shoes;  
table region / list out=region_freq1;  
run;  
  
proc freq data = sashelp.shoes;  
Table region*product / list out=region_freq2;  
run;  
  
proc freq data = sashelp.shoes;  
table region*stores / list out=region_freq3;  
run;
```

Not only use for
data processing,
also use for
procedures

4.

Limit the number of times you sort your data

```
data new;  
  infile 'rawdata.dat';  
  input ID $ 1-4 name $ 5-25 salary 26-35;  
run;  
  
proc sort data=new out=new_sorted presorted;  
  by ID;  
run;
```

If you think the incoming data is already sorted, use the **presorted** option on your SORT statement; the sort order will be verified.

4.

Limit the number of times you sort your data

When creating an SQL view, avoid including an ORDER BY clause in the view, as the data will need to be sorted every time the view is used.

```
proc sql;  
  create view sql.new as  
  select *  
  from sql_old  
  order by firstvar;  
  
proc print data=sql.new;  
Run;
```

The PROC PRINT or any other procedure/Data step that uses the view will execute the stored SQL query, including the ORDER BY.

4.

Overview of SQL views

A PROC SQL view contains a stored query that is executed when you use the view in a SAS procedure or DATA step. Views are useful for the following reasons:

- Often save space, because a view is frequently quite small compared with the data that it accesses
- Shield sensitive or confidential columns from users while enabling the same users to view other columns in the same table
- Ensure that input data sets are always current, because data is derived from tables at execution time
- Hide complex joins or queries from users

4.

When is a sort required

Requires sorting

- DATA step with SET or MERGE and BY statements
- By statement in PROC MEANS, PROC FREQ, etc.

Does not require sorting

- PROC SQL joins (unless it includes an ORDER BY statement)
- CLASS statements in PROC MEANS, PROC FREQ, etc

5.

Select only the columns you need when working with SAS data

Do This:

```
data new;  
  set old (drop=category  
           type value ...);  
  more statements here;  
run;
```

Not This:

```
data new;  
  set old;  
  more statements here;  
run;
```

Variations:

- Use the keep= option if you need to keep less variables than you need to drop!
- Use both keep= and drop= options to control variables on both the incoming and outgoing sides!
- Keep= and drop= options can be used in PROC steps, too!

6.

Select only the rows you need when working with SAS data

Do this:

```
data new;  
  infile 'old.dat';  
  if city='CLEVELAND';  
  more statements here;  
run;
```

Not this:

```
data new;  
  infile 'old.dat';  
  more statements here;  
run;
```

6.

Consider the position of the subsetting IF

Do this:

```
data new;  
  infile 'old.dat';  
  if city='CLEVELAND';  
  more statements here;  
run;
```

Not this:

```
data new;  
  infile 'old.dat';  
  more statements here;  
  if city='CLEVELAND';  
run;
```

Subset as soon as you have all necessary values in order to prevent unnecessary creation of variables and additional processing.

7.

If you are reading SAS data, use WHERE instead of subsetting IF

Try this:

```
data new;  
  set old;  
  where condition;  
  more statements here;  
run;
```

Instead of this:

```
data new;  
  set old;  
  if condition;  
  more statements here;  
run;
```

WHERE is a pre-processor. It subsets data before it is loaded into the Program Data Vector (PDV).

Added efficiency: when using SAS/Access engines, SAS attempts to send the WHERE clause to the RDBMS for evaluation rather than to SAS;

With the IF statement, SAS must do the processing.

The Program Data Vector

- The PDV is a place where SAS stores lines of data in memory before writing an observation to a dataset on the disk.
- Created during compile phase, filled in with one line of data at a time during execution.

_N	_ERROR	ID	Na me	Addre s	Height	Weight	Age
.

7.

If you're going to run a procedure on the data, use the "where" statement in the procedure

Instead of this:

```
data new;  
  set old;  
  where condition;  
run;  
proc means data=new;  
  more statements  
  here;  
Run;
```

Try this:

```
proc means data=old;  
  where condition;  
  more statements here;  
run;
```

8.

Consider declaring variables as character when there is a storage savings.

Consider Employee ID values like the following:

```
1015  
2034  
5543  
6793  
...
```

```
data new;
```

```
input ID 1-4;
```

- ID is numeric requiring 8 bytes of storage

```
data new;
```

```
input ID $ 1-4;
```

- ID is character requiring 4 bytes of storage

A savings of 4 bytes per observation adds up when dealing with large data!

9.

Use Built in Features and Functions

Better:

```
if (upcase(a) = 'YES') then x = 1;
```

When testing for all possible combinations

Works:

```
if (a = 'YES' or  
    a = 'YEs' or  
    a = 'YeS' or  
    a = 'yES' or  
    a = 'yeS' or  
    a = 'yEs' or  
    a = 'Yes' or  
    a = 'yes' ) then x = 1;
```

9.

Use Built in Features and Functions

Types of Functions

- Character (SUBSTR, LEFT, RIGHT, UPCASE)
- Arithmetic (ABS, SUM, SQRT)
- Array (DIM)
- Date and Time (TODAY, YRDIFF, MDY, TIMEPART)
- Financial (MORT, NPV, SAVINGS)
- Mathematical (LOG, EXP)
- Probability (POISSON, PROBCHI)
- Quantile, Random Number (NOMINAL, UNIFORM)
- Sample Statistics (MEAN, MIN, MAX, STD, NMISS)
- Special (LAG, PUT, INPUT)

[Introduction to SAS Functions Paper](#)
[SAS 9.4 Language Reference Documentation](#)



10.

Use the Correct Tools

- NOT a DATA STEP
 - Must process each row in the table
 - Creates another copy of the table

```
data sasdata.ManyColumns;  
  set sasdata.ManyColumns;  
  format date weekdate18.  
          balance dollar 12.2;  
  label cust_id = 'Customer ID';  
run;
```

- Use DATASETS to assign column attributes
 - Formats
 - Labels

```
proc datasets lib=sasdata nolist;  
  modify ManyColumns;  
  format date weekdate18.  
          balance dollar 12.2;  
  label cust_id = 'Customer ID';  
quit;
```

10.

Use the Correct Tools

Do not recode

```
data two;
  set one;

  if age lt 18 then
    age_group='Minor';
  else if age lt 30 then
    age_group='Young Adult';
  else if age lt 45 then
    age_group='Early Family';
  else if age lt 60 then
    age_group='Middle Age';
  else age_group='Aged to Perfection';
```

Use Formats

```
value agerange
  Low - <18      = 'Minor'
  18  - <30     = 'Young Adult'
  30  - <45     = 'Early Family'
  45  - <60     = 'Middle Age'
  60  - high    = 'Aged to Perfection';
```

10.

Use the Correct Tools

Use SAS Date Values

- Valid from 1582 A.D. to 1990 A.D.
- Many formats available for grouping/reporting

```
date.          11JUL18
date9.         11JUL2018
worddate.      July 11, 2018
weekdate.      Wednesday, July 11, 2018
month.         7
monname.       July
monname3.     Jul
monyy7.       JUL2018
```

SAS Dates

What are they?

- The SAS System represents dates as the number of days since a reference date.
- The reference date, or date zero, used for SAS date values is **1 January 1960**.
- For Example

January 1, 1960	0
December 31, 1959	-1
February 3, 1960	33
October 17, 1991	11612
March 30, 2022	22734

Resources

Where to learn more?

20 Ways to run your SAS Programs Faster

Video and Paper



SAS® GLOBAL FORUM 2020

Twenty Ways to run your SAS program faster and use less space

Stephen Sloan, Accenture

Stephen, a Data Science Senior Principal at Accenture, leads multi-functional projects has presented at over 20 SAS events and been published in professional journals.

Stephen has a B.A. cum laude, Honor in Mathematics, Brandeis University, M.S. in Mathematics and Computer Science, Northern Illinois University, MBA (1st in class), Stern Business School (New York University), and a graduate certificate in Financial Analytics from Stevens Institute.

USERS PROGRAM SAS GLOBAL FORUM 2020

0:25 / 20:00

[SAS Global Forum 2020 – Video](#)

[SAS Global Forum 2020 - Paper](#)

Resources Online

SAS Tutorials on Programming

The screenshot displays the SAS Online Resources interface. On the left is a navigation menu with categories such as 'Products & Solutions', 'Analytics in Action', 'How To Tutorials', and 'Programming'. A blue arrow points to the 'Programming' category, which is expanded to show sub-items like 'Getting Started', 'Risk Management', 'SAS Studio', and 'SAS Viya'. On the right, a grid of 12 tutorial thumbnails is shown, each with a title and a duration. The thumbnails include:

- Getting Started with SAS Studio (9:18)
- Writing a Basic SAS Program (6:14)
- Getting Started with SAS Windowing Environment (6:44)
- Using SAS User Interfaces (3:05)
- Working in SAS Studio (7:06)
- SAS Programming Tip: Send E-mail with Attachments (6:15)
- Encoding: The Key to Handling Multilingual SAS® Data from All over the World (13:11)
- SAS Programming Tip: Setting System E-mail Options (5:24)
- SAS Programming Tip: Write an E-mail But Delay Sending (3:09)

Resources

SAS Global Forum Papers

- [Leave Your Bad Code Behind: 50 Ways to Make Your SAS® Code Execute More Efficiently](#) William E Benjamin Jr, Owl Computer Consultancy, LLC
- [SAS® Shorts: Valuable Tips for Everyday Programming](#) Jeff McCartney and Raymond Hu, Social and Scientific Systems, Inc., Bethesda, MD
- [Productivity Tips for SAS® Enterprise Guide® Users](#) Jennifer First and Steven First, Systems Seminar Consultants, Madison, WI, United States
- [Tips and Techniques for the SAS® Programmer](#) Helen Carey, Carey Consulting, Kaneohe, HI, Ginger Carey, Carey Consulting, Kaneohe, HI

To search past papers
[click here](#)

Q&A

Please submit your questions using the Q&A icon located in the menu at the bottom of your screen

