

# 180 Terrific Ideas for Using and Learning the SAS® System

Helen Carey, University of Hawaii Computing Center  
Ginger Carey, University of Hawaii Computing Center



## Introduction

Learning and using the SAS System can be both exciting and challenging. To keep current with the latest features and enhancements to SAS is an ongoing process. Besides keeping up-to-date, there is also the need to be efficient and to get the desired results on time.

For over 15 years, we have been working with SAS users in a university environment. Many have excellent work habits and use SAS well; others have a lot to learn. As we help each one, we find many tips being repeated over and over again.

Perhaps you have seen *Life's Little Instruction Book* by H. Jackson Brown, Jr., which has "511 suggestions, observations, and reminders on how to live a happy and rewarding life." Well, it's time for us SAS enthusiasts to have our own list of both new ideas and reminders of how to be more productive and happy. There is a wealth of information available from SAS publications, SAS users, experience and the SAS-L electronic discussion group.

Why 180 ideas? Simply because this is the 18th annual SUGI Conference. Each idea presented is concise. The list of ideas runs the gamut from daily work habits to annually attending the SUGI Conference. Remember that with SAS, there are a number of ways to accomplish your goal. These ideas are based on using SAS 6.07 on the IBM mainframe. Choose the ideas that will work for your situation and style. You may not agree with every idea but these are ones that have been helpful to us.

## Learning More

1. **Attend the SUGI Conference.** Enough said. You are here.
2. **Go early for a seat at the popular presentations at SUGI.** Presentations by Jim Goodnight and Dave Brumitt are well attended; so are popular topics, such as new features and the tutorials.
3. **Go to the SUGI Opening Session.** If you didn't make it this year, go next year. You'll learn about the future directions of the SAS System and have fun.
4. **Take business cards to SUGI.** Today, business cards need to have your FAX number, your E-mail address, and all of the other items, like your name.
5. **Visit the demo area at SUGI,** preferably before Wednesday morning.
6. **Watch the SAS one-hour videos —** they are worthwhile. The videos are a cost item but if you have money in your training budget, buy them.
7. **Attend the training offered before the Conference —** it's excellent. A cost item, but we believe in training.
8. **Watch only one hour maximum of a video at a time.** You will remember more later. With marathon viewing, you will remember that you can do it in SAS, but you won't remember how. Then it's back to the video.
9. **Teach those who use obsolete methods the new way or they will teach others the old way.** It is painful to watch someone use a clumsy, unfriendly line editor. Teach them new ways or hope that they retire soon. If you don't know what a line editor is, then you probably don't know the origin of the CARDS; statement.
10. **Learn by doing.** Experience is a good teacher.
11. **Learn by teaching.** Help your colleagues and your SAS users group. To teach is to learn.
12. **Learn about the new features and enhancements.**
13. **Learn macros well.** Macros make your programming more efficient and powerful. However, they can be tricky.
14. **Learn to create data views in the data step or in PROC SQL** rather than keeping many subsets of your data set.
15. **Learn PROC SQL.** You can access data, do complex merging and do subsetting with it.
16. **Learn PROC REPORT.** You can interactively design your report and keep the definition to use again.

17. **Remember you can't learn less.** Take comfort in the fact that with each programming experience, you are learning something, even from your mistakes.
18. **KEEP LEARNING.** You really don't have a choice — not with all the great new features in each new release.
42. **Use the FIND command with the ICASE option to ignore case.**
43. **Try to end the unbalanced quotation marks** which you usually have when you get the message "... quoted string has become more than 200 characters ..." by submitting \*'; \*"; RUN;  
RECALL, fix your program and re-submit.

## Work Habits

19. **Keep an open mind.** Approach the problem with the attitude that *it CAN be done in SAS.*
20. **Use the right tools for the job.** Don't try to do everything in SAS. Hand calculators work well even though SAS has the DCALC window.
21. **Plan ahead.**
22. **Write down your plan.**
23. **Think creatively.**
24. **Ask.** Don't assume that you know a client's needs.
25. **Listen.**
26. **Anticipate problems.** Remember Murphy's Law—"If anything can go wrong, it will."
27. **Divide the project into manageable pieces.**
28. **Reward yourself** when you complete one of those manageable pieces. A pat on the back, a trip to Hawaii (the latter was brought to you by the Hawaii Visitors' Bureau).
29. **Follow-up on a project** with a telephone call.
30. **Don't spend 1 hour to save 10 minutes.**
31. **Simplify.**
32. **Organize.**
33. **Keep a journal.** Use paper to store information, instead of relying on your memory.
34. **Take breaks.** Don't overdose on SAS.
35. **Get rid of clutter.** Do your housekeeping.
36. **Backup your files.**
37. **Keep your backups elsewhere.** The burglar stealing your computer also may want the diskettes next to it.
38. **Work smarter, not harder.**
44. **Clean your windows.** Or should that be clear your windows, the LOG and OUTPUT, before re-submitting the program.
45. **Close windows you no longer need.**
46. **Use the NOTEPAD window for notes to yourself.**
47. **Use the NOTEPAD window for viewing another program.** Type NUMS ON to get the line numbers.
48. **Use the LIBNAME window and the subsequent DIR, VAR, and FSVIEW windows.**
49. **Have your organization acquire the SAS/FSP® product.** Then you can use the FSVIEW and FSEDIT windows from the display manager.
50. **Set the function keys to the way you work.**
51. **Set the function keys the same on all platforms you use.**
52. **Put :ts on a function key** to easily split the text at the cursor position.
53. **Put ? on a function key** to easily recall previous commands.
54. **Keep the key layout with your changes by your keyboard.**
55. **Get a mouse.**
56. **Use the mouse.** You can do it; it just takes practice.
57. **Learn to cut and paste with your function keys and mouse.**
58. **Save the source code** in the same SAS library as the data file by using the SAVE command, such as  
SAVE libref.catlg.name.SOURCE.
59. **Use the UNDO function key.**
60. **Use the SUBTOP command** to submit the top lines in the PGM window.
61. **Cut and paste from windows that do not support the PRINT command** to the PGM window or NOTEPAD window that does. This is the way to print your key definitions that can be viewed but not printed from the KEYS window.

## SAS Display Manager System

39. **Use the SAS Display Manager System.**
40. **Read the log out loud** until you are able to read it with understanding. Track down all bugs and pay attention to the notes and warnings.
41. **Use the RUN; statement** between DATA and PROC steps.
62. **Set AUTOSCROLL MAX or AUTOPOP OFF** in the OUTPUT window to enable execution, without suspension, of all procedures submitted.
63. **Use the WSAVE command** to save your window setting changes, such as color, location or AUTOPOP OFF.

64. Set your PAGESIZE option to 60 or 65 and your LINESIZE option to 80 or 132 if you plan to print to the printer.
65. Reset the page number before your final run for printing. Use  

```
OPTIONS PAGENO = 1;
```

 If your printout begins with page 238, it gives away the number of times you ran the program before printing and using white-out to cover-up is not the best way.
66. FREE your print file before issuing the PRINT command from another window.
67. Use the NEXTSCR and PREVSCR commands to move through the multiple screens displayed in the FSFORMS window.
68. Use the HELP window.
69. Learn to navigate around the HELP window by using it. It gets easier the more you use it.
70. Use =x to immediately end the HELP window.
71. Use the SAS Sample Library. You may want to run the index program in the SAS Sample Library.
72. Customize the display manager for your work habits. You can always delete catalog entries for windows, forms, and key definitions that you customized and no longer use.

## Testing and Debugging

73. Develop and test large applications incrementally.
74. Insert an ENDSAS; statement after the first part of the program that you want to test, rather than commenting out the rest of the program when running in batch mode.
75. Use RUN CANCEL; to compile and check syntax without executing. Global statements, such as OPTIONS will still run. *Data steps with internal data lines will still run.*
76. Use the /\* and \*/ delimiter comments to block out code and prevent it from executing.
77. Use a subset of your data for testing. Take a random sample if the data is organized in some order. See the *SAS Language and Procedures: Usage 2, Version 6, First Edition* for examples of sampling.
78. Ask yourself what changed when your program no longer works and it worked before. Something had to have changed; it's different data or even a different day. The basic concept in the book *The Rational Manager* is to discover what changed to create the problem.
79. Use the Usage Notes.

80. Update the Usage Notes quarterly. Ask your organization to request the latest copy.
81. Take your work, printouts and diskettes with files, with you when seeking help.
82. Take your work with you when seeking help. It's worth repeating.
83. Walk and think or even read the manual when you get stuck. This is often more productive than a continuous loop of making changes and re-running the program without fully understanding why you made the change.

## Programming Style

84. Use meaningful data set names.
85. Use meaningful variable names.
86. Start temporary variable names with an underscore. Underscores can also be used in scratch data set names. They make it easier to find the temporary variables and scratch data sets for dropping or deleting later.
87. Label your variables.
88. Use program headers. This is a block of standardized comments at the beginning of your program, which usually includes the program name, the programmer's name, date written and revised.
89. Use mixed case for your programming statements. It is easier to read. We use uppercase for SAS words and lowercase for names we construct.
90. Use indentation to reflect logic and structure.
91. Use blank lines for clarity.
92. Make your SAS program readable.
93. Comment, comment, comment.
94. Document, document, document.
95. Use the SAS program written by Steve James of the Centers for Disease Control to document SAS data sets. You can get the program from the SAS-L archives by searching for 'Steve James documenting'.
96. Document by always specifying the SAS data set instead of letting SAS default to the last data set created. If you insert another data step, there will be no surprises.
97. Avoid clever coding techniques that saves a micro-second of machine time and waste one hour of your time trying to remember what you did. If you need to be clever, comment what you did.
98. End user-defined format names with FMT.
99. Keep your PROC FORMAT programming statements in a separate file.

100. Learn to use the `FILENAME` and `LIBNAME` statements. For IBM/MVS programmers, this is better than using Job Control Language.
101. Use the `IN` operator, such as  

```
IF grade IN (3,6,9,12) ... ;
```

 It's easier to write, read, maintain, and change.
102. Use the condensed form for comparisons linked by `AND` which have a common variable. It is easier to understand,  

```
IF 100 <= y < 300 ...
```

 versus  

```
IF 100 <= y AND IF y < 300 ...
```
103. Use `TITLE` statements to identify the report and project. Add secondary titles or footnotes to further explain the data analysis. Years or days later, it will help you remember the reason for the report.
114. Use double hyphens to choose variables stored consecutively. This saves typing, i.e.  

```
gender--ethnic;
```
115. Use the single hyphen to choose variables with the same prefix, i.e.  

```
class1-class19 ;
```
116. Use the fact that the `=` (equal) operator returns the value 0 (false) or 1 (true). For example, to count the number of questions with the answer 3, you can use  

```
TOTAL=SUM(q1=3,q2=3,q3=3,q4=3);
```
117. Specify another format library or different search order by using  

```
OPTIONS FMTSEARCH=(lib1 WORK);
```
118. Use double quotes to enclose text containing macro variable references. For example,  

```
TITLE "Today is &SYSDATE";
```
119. Know the difference between the `NODUPKEY` and `NODUPPLICATES` on `PROC SORT` to eliminate duplicate records.
120. Store the information that your data set is already sorted by using the `SORTEDBY=` data set option. For example:  

```
DATA survey(SORTEDBY=dept);
```

## Programming and Efficiency

104. Know which resources are important to you — your time, computer time, computer memory, speed, or storage space.
105. Don't re-invent the wheel or SAS code.
106. Look at statements and techniques with an open mind and do not limit yourself to the same old way. For example, use a `FORMAT` statement in a `PROC` step to select the area code from the phone number instead of the `SUBSTR` function in a data step. For example:  

```
PROC FREQ DATA=survey;
  TABLE phone;
  FORMAT phone $3.;
```
107. Use the `*` and `*/` delimiter comments. They are more efficient than the `** ;` statement comments. This is due to the manner in which they are parsed by the supervisor.
108. Print out the `BY` variable information in your reports by using `#BYVAR1`, `#BYLINE1`, and `#BYVAL1` in the `TITLE` statement.
109. Use explicit conversions, not implicit. You'll know exactly what SAS is doing and when you see conversion notes in the `LOG`, you will know to check it out.
110. Use the `WHERE` statement. It's powerful. However, be aware that `WHERE` statements are not cumulative unless you use the `WHERE ALSO` statement.
111. Understand the data vector if you do `SETS` and `MERGES`.
112. Understand dates — date variables, date formats and how date values are stored.
113. Review the available functions. They are useful. Carefully read about how the function works.
121. Remember **GIGO** — garbage in, garbage out. Sometimes the hardest, most time-consuming and least fun part of the project is cleaning up the data.
122. Examine your data. Taking a few minutes to examine your data can save many hours and trees later. Use frequency counts, means, minimum, maximum, plots, etc. to find bad data.
123. Use `PROC PRINT` to check the results of data sets that you create, even if it is just to print the first few observations. For example to print the first 20 observations, use  

```
PROC PRINT DATA=survey(OBS=20);
```
124. Print and check the variable information by using  

```
PROC CONTENTS DATA=aloha POSITION;
```

 Scan the listing to check if the variables and their attributes are correct and complete.
125. Keep all variables and don't write over existing variables until you are certain that the data set looks correct.
126. Read and keep only the variables that are needed.
127. Read an external file only once.
128. Create multiple SAS data sets within one `DATA` step by using the `OUTPUT` statement.

## Your Data

129. **Create an index where appropriate.** See the manual *SAS Programming Tips: A Guide to Efficient SAS Processing*. Indexes improve search efficiency in FSEDIT when the WHERE command is used.
130. **Check test results against known results.**
131. **Know your data.** Mistakenly, we threw out physical activity whose duration was longer than 4 hours. However, gardening was a choice for exercise and some retired people in Hawaii garden for 8 hours a day.
132. **Create user-defined informats and formats to code, decode, and recode data.**
133. **Use the MISCOVER or TRUNCOVER option on the INFILE statement if you may have missing data at the end of your list input.**
134. **Use the DSD option on the INFILE statement if you are reading comma delimited files, such as those from dBASE. It allows commas within quoted strings. The statement would look like**  

```
INFILE myfile DSD MISCOVER;
```

*(By the way what does DSD stand for?)*
135. **Know that SAS strips leading blanks when reading and/or printing character data unless you use a \$CHAR format.**
136. **Save only the final SAS data set in a permanent SAS library. Make all the others temporary.**
137. **Remember that SAS 6.07 can create a SAS 5.18 library.**
138. **Use PROC DATASETS to delete a temporary dataset in the middle of a program when you are running a very large job and need the disk space.**
139. **Calculate the size of a SAS data set by using the formula:**  

$$\text{size in bytes} = 218 + (\# \text{ variables} * 106) + (\# \text{ obs} * (4 + \text{sum of the length of all variables})).$$
140. **Use the LENGTH statement before the INPUT or SET statement to reduce the data storage space for SAS variables.**
141. **Use the LENGTH statement to specify a character variable's length explicitly.**
142. **Understand the effects of truncation before using the LENGTH statement for your numeric variables. Refer to the length table in your host system companion manual. If you choose too short a length for the variable's precision, a number that you thought was being stored correctly isn't and SAS doesn't tell you.**
143. **Store classification variables containing numerical values as character variables to save storage space.**

## Resources

144. **Use your SAS manuals.**
145. **Write notes in your manuals and in the index.** It's OK to write in your book.
146. **Add tabs or bookmarks to the functions or other sections of a manual that you reference frequently.**
147. **Use the SUGI Proceedings.** They are full of excellent papers, ideas, and codes. Take advantage of them. Don't re-invent the wheel.
148. **Buy the index to the SUGI Proceedings.**
149. **Buy the index to SAS documentation.**
150. **Subscribe to Observations™.** This one costs.
151. **Subscribe to SAS Communications®.** This one is free.
152. **Get the SAS Communications Index.** Send to LISTSERV@VM.SAS.COM the commands:  

```
GET SASCOMM1 ANNOUNCE
GET SASCOMM2 ANNOUNCE
```

or get the file from the anonymous FTP service at site FTP.SAS.COM. Technical Support files are located in the subdirectory techsup.
153. **Buy the manual SAS Programming Tips: A Guide to Efficient SAS Processing.** It offers practical examples and guidelines.
154. **Read the Changes and Enhancements technical reports.** We found out about the DSD option and it was exactly what we needed to read a dBASE delimited file.
155. **Review the SAS Companion for your operating system.**
156. **RTFM.** Read the [friendly] manual. This is such an important concept that the SAS-L group sold T-shirts with RTFM on the back.
157. **Read the manual, if all else fails.** Well, it's important.
158. **Find out about the user written SAS publications such as Professional SAS® Programming Secrets by Rick Aster and Rhena Seidman.**
159. **Buy a bigger bookcase.**
160. **Find someone to give you A User's Guide to SAS 76 manual.** It's a good reminder of simpler days; it's probably a collector's item. Therefore, go to your bookcase, remove that old obsolete SAS 76 manual and mail it to us.
161. **Join a SAS Users Group in your area and contribute to the group.**
162. **Join SAS-L, the electronic discussion group.** You can join it via BITNET, INTERNET, or MCI mail, Compuserve, and other E-mail systems.
163. **Learn how to search the SAS-L Archives.**

## Beginners' Mistakes

164. Mind your s's and q's - semicolons and mismatched quote marks.
165. Use a period when checking for missing numeric data and use a blank enclosed in quotes when checking for missing character data. For example:  
`IF age = . OR ethnic = ' ' ... ;`
166. Create one format for variables coded the same and apply it to each of the variables. Don't make the mistake of creating identical user-defined formats with different names.
167. Keep in mind that some procedures, such as PROC NPAR1WAY or PROC FREQ, only use the first 8 or 16 characters even though you can store 200 characters in the format values. If different group values truncate to the same 8 characters, then they will be grouped together.
168. Recode a variable by creating a user-defined format and using the PUT function instead of using many IF statements. For example:  
`dept=PUT(deptno, deptfmt.);`
169. Use the SET statement or a DATA= option on the procedure statement to use an already created data set. You don't need to rerun the data step that reads the raw data.
170. Use the SUM function for finding totals when you want to ignore the missing values. For example, use:  
`expense=SUM(cash, check, charge);`  
Otherwise, if CASH is missing then the results for EXPENSE will be missing if you use:  
`expense = cash+check+charge;`
171. Use the CLASS statement with procedures where possible rather than sorting and using the BY statement.
172. Use the MISSING option on the PROC TABULATE statement to include all observations in the tables. Otherwise, observations with a missing value for any of the variables on the CLASS statement will be omitted.
173. Use the MISSING option on the FREQ procedure TABLES statement if you have grouped missing and non-missing values together using PROC FORMAT and you want this group included in the tables and statistics.
174. Use the UPCASE function to do comparisons on character variables when it is not known if the data is in uppercase or lowercase.
175. Use the LABEL option on PROC PRINT to print out variable labels as column headings.

176. Remember that SAS is SAS. SAS is no longer an acronym for Statistical Analysis System; it's just SAS.
177. Pronounce SAS as "sas", which rhymes with "lass", not as S - A - S.
178. Know that SAS is an Information Delivery System. Remember that SAS does a lot more than statistics.
179. Be happy that you use such a powerful system.
180. *Live and Learn and Pass It On.* This is our final idea and, it seems to us, the atmosphere of the SUGI conference.

This year 180 terrific ideas. Now we're ready to increase it to 365. Why? Wouldn't you like to have a daily desk calendar with SAS tips and techniques to pass on?

## Acknowledgments

We thank Christopher Chi-Chung Ma for preparing the slides for this presentation.

SAS, SAS/FSP, SAS Communications, and Observations are registered trademarks or trademarks of the SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

## References

- Brown, H. Jackson, Jr. (1991), *Life's Little Instruction Book*, Nashville, Tennessee: Rutledge Hill Press.
- Brown, H. Jackson, Jr. (1992), *Live and Learn and Pass It on*, Nashville, Tennessee: Rutledge Hill Press.
- Kepner, Charles H. and Tregoe, Benjamin B. (1965) *The Rational Manager: A Systematic Approach to Problem Solving and Decision Making*, New York: McGraw Hill Book Company.

## Authors Contact

Send your favorite ideas to:

Helen Carey CAREY@UHCCMVS.BITNET  
Ginger Carey GINGER@UHCCMVS.BITNET  
GINGER@UHCCMVS.UHCC.HAWAII.EDU

University of Hawaii Computing Center  
2565 The Mall, Keller Hall  
Honolulu, HI 96822

phone: (808) 956-2387 fax: (808) 956-2412