PHUSE EU Connect 2021

Paper CT08

Differences in performance between the options KEEP and WHERE as compared to the respective statements KEEP and IF in SAS data steps.

Ruurd Bennink, OCS Life Sciences, 's-Hertogenbosch, the Netherlands

ABSTRACT

In general, it is assumed that keeping variables while reading in a SAS dataset (via the KEEP option on the SET statement) is more efficient than applying a KEEP statement within the SAS data step.
It is also assumed that selecting observations while reading in a SAS dataset (via the WHERE option on the SET statement) is more efficient than applying the IF statement.
This paper will look into the difference in performance of using the beforementioned options and statements by applying them in several scenarios. Is there really a big difference in performance? Or is it negligible?

INTRODUCTION

What is performance and how is it measured?

Within this paper, the performance of a SAS program is determined by the time you need to wait until your program run is finished. This time is given in the SAS log file as 'real time'. The longer the 'real time' is, the worst the performance is. In the SAS log file the 'real time' is presented after the conduct of each data step or SAS procedure.

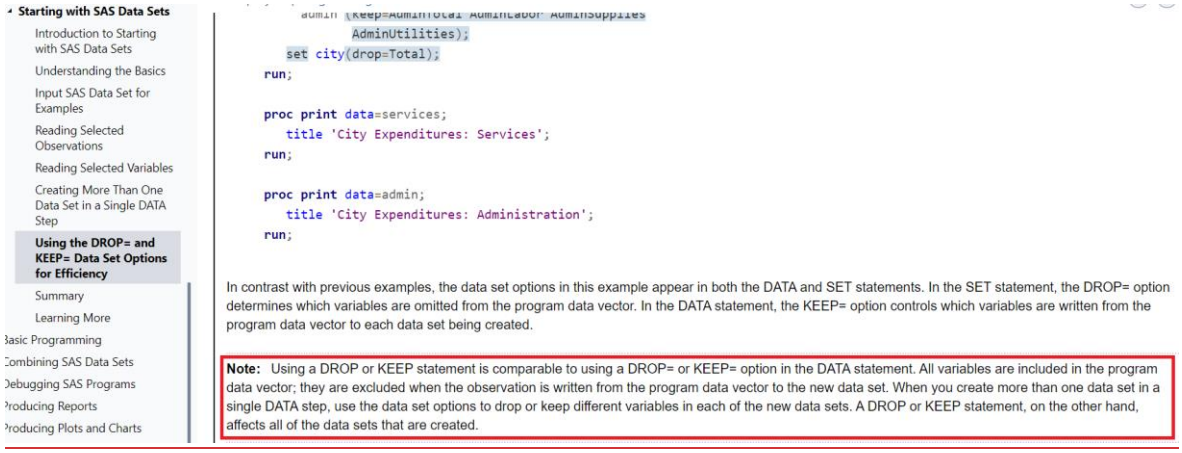
Background

SAS literature gives [1], among many others, two tips to increase the performance of your SAS programs:

1) Use the KEEP or DROP option instead of the KEEP or DROP statement in a data step.

Using the KEEP option on the SET statement means that only the selected variables are included in the program data vector (PDV). On the other hand, when using the KEEP statement all variables will be included into the PDV. Hence it is expected that using the KEEP option on the SET statement will show a gain in performance as compared to the KEEP statement. Another possibility is to use the KEEP option on the DATA statement, however that will have the same effect as using the KEEP statement as indicated in the SAS

documentation on DROP and KEEP efficiency:



2) Use the WHERE option or statement instead of the IF statement in a data step.

From the SAS documentation it is also known that the WHERE option on the SET statement is more efficient than using the IF statement.

When using the WHERE option on the SET statement, only the selected observations will be included in the Program Data Vector (PDV) whereas the IF statement will include all observations in the PDV. Another option would be to use the WHERE statement, but this has the same effect as the WHERE option on the SET statement.

It is examined in this paper to what extent these two tips contribute to the improvement of performance.

THE EFFECT OF USING THE KEEP OPTION INSTEAD OF THE KEEP STATEMENT

To examine this effect in detail a dataset was simulated with 50 randomly filled variables, 25 character variables and 25 numeric variables, and one million observations. This dataset is then tested by keeping only two variables.

```
DATA all;

  ARRAY cvar{25} $25;
  ARRAY nvar(25) 8;

  DO vari=1 to 25;
    DO chari=1 to 25;
       substr(cvar{vari},chari,1)=byte(floor(ranuni(354)*26)+62);
    END;
  END;

  DO vari=1 to 25;
    nvar{vari}=ranuni(354);
  END;

  DO obsi=1 to 1000000; OUTPUT; end;

RUN;

%macro keeploop;

  %do i=1 %to 25;

    %put KEEP statement;
```

```
    DATA var10_1;
      SET all;
      KEEP cvar10 nvar10;
    RUN;

    %put KEEP option;
    DATA var10_1;
      SET all (keep=cvar10 nvar10);
    RUN;

  %end;

%mend keeploop;

%keeploop

/* End of Program */
```

Example: the SAS log shows that the real times for both the KEEP option and KEEP statement are quite small:

KEEP statement

```
NOTE: There were 1000000 observations read from the data set WORK.ALL.
NOTE: The data set WORK.VAR10_1 has 1000000 observations and 2 variables.
NOTE: DATA statement used (Total process time):
      real time          0.38 seconds
      cpu time           0.37 seconds
```

KEEP option

```
NOTE: There were 1000000 observations read from the data set WORK.ALL.
NOTE: The data set WORK.VAR10_1 has 1000000 observations and 2 variables.
NOTE: DATA statement used (Total process time):
      real time          0.35 seconds
      cpu time           0.36 seconds
```

The above snippet from the log file presents only one run with KEEP option and KEEP statement. Because the performance will fluctuate due to interfering job runs of other users on the server throughout the day it is necessary to perform several runs. The macro KEEPLOOP in the program above performs this cycle 25 times. After those 25 runs it is possible to make a robust statement about the difference in performance between KEEP option and KEEP statement.

A SAS program was created that reads in the log file and makes a comparison between the real times of the 25 KEEP statements vs. the real times of the corresponding 25 KEEP options.

Four scenarios were used (A1, A2, A3 and A4) to examine the difference in performance between the use of the KEEP option and the KEEP statement. Scenario A1 is presented below in detail. The other three scenarios are presented in summary in Table 1.

The percentage increase in performance of the KEEP option vs. the KEEP statement is calculated as:

100(real time KEEP statement – real time KEEP option)/(real time KEEP statement) (%)*

Scenario A1: A simulated dataset with 50 variables (25 character and 25 numeric) and one million observations with two variables to keep.

The resulted output from the program is shown below, with yellow marking the numbers corresponding to the earlier log snippet:

KEEP statement vs. KEEP option, 50 vars of which 25 numeric and 25 character, 1.000.000 observations

Obs	Real time used for KEEP statement (s)	Real time used for KEEP option (s)	Percentage increase in performance Option vs statement
1	0.38	0.35	7.8947
2	0.36	0.35	2.7778
3	0.35	0.34	2.8571
4	0.37	0.34	8.1081
5	0.36	0.35	2.7778
6	0.36	0.34	5.5556
7	0.38	0.35	7.8947
8	0.36	0.34	5.5556
9	0.36	0.34	5.5556
10	0.36	0.34	5.5556
11	0.35	0.34	2.8571
12	0.37	0.35	5.4054
13	0.36	0.34	5.5556
14	0.37	0.35	5.4054
15	0.36	0.35	2.7778
16	0.36	0.35	2.7778
17	0.39	0.35	10.2564
18	0.35	0.34	2.8571
19	0.36	0.36	0.0000
20	0.36	0.35	2.7778
21	0.36	0.34	5.5556
22	0.36	0.35	2.7778
23	0.36	0.34	5.5556
24	0.36	0.34	5.5556
25	0.35	0.35	0.0000

The MEANS Procedure

Analysis Variable : perc_dif Percentage increase in performance Option vs statement				
N	Mean	Std Dev	Minimum	Maximum
25	4.5858937	2.4661464	0	10.2564103

The above table shows that the use of the KEEP option is about 4.6% more efficient, on average, after 25 runs with a minimum of 0% and a maximum of 10.3% and a Standard Deviation of about 2.5%.

Scenario A2: A simulated dataset, also with one million observations, but with 200 randomly filled variables, 100 character variables and 100 numeric variables, also keeping 2 variables.

Scenario A3: A simulated dataset, also with one million observations, but with 400 randomly filled variables, 200 character variables and 200 numeric variables, also keeping 2 variables.

Scenario A4: A simulated dataset, also with one million observations, but with 200 randomly filled variables, 100 character variables and 100 numeric variables, but keeping 1 variable instead of 2.

Table 1 below shows an overview of the difference in performance (%) for each of the four scenarios.

Table 1: Performance gain of using KEEP option instead of KEEP statement after 25 runs.

Scenario	N_{keep}: N_{var}	Performance gain	
		Mean (%)	SD (%)
A1	2:50	4.6%	2.5%
A2	2:200	6.5%	1.5%
A3	2:400	17.5%	2.1%
A4	1:200	12.8%	2.8%

SD = Standard Deviation, N_{keep} is the number of variables kept and N_{var} is the number of variables in the input dataset.

In conclusion, based on the four scenarios, the gain in performance of using KEEP option instead of KEEP statement increases when the number of variables kept gets smaller as compared to the number of variables in the input dataset.

THE EFFECT OF USING THE WHERE OPTION INSTEAD OF THE IF STATEMENT

To examine the difference in performance between the use of the WHERE option on the SET statement instead of the IF statement in a SAS data step a similar approach was used as for the comparison between the performance of the KEEP option on the SET statement vs. the KEEP statement. Thereby noted that the WHERE statement works the same as the WHERE option on the SET statement. Almost the same simulated dataset was used (25 character variables, 25 numeric variables, 10 million observations), but including a random number between 0 and 1 to each observation (variable EXTRAVAR) to select approximately 50% of the records:

```
DATA all;

  ARRAY cvar{25} $25;
  ARRAY nvar(25) 8;

  DO vari=1 to 25;
    DO chari=1 to 25;
      substr(cvar{vari},chari,1)=byte(floor(ranuni(354)*26)+62);
    END;
  END;

  DO vari=1 to 25;
    nvar{vari}=ranuni(354);
  END;

  DO obsi=1 to 10000000;
    extravar=Ranuni(354);
    OUTPUT;
  END;

RUN;

%macro keeploop;

  %do i=1 %to 25;
    %put IF statement;
    DATA var10_1;
      SET all;
      IF extravar > 0.5;
  RUN;

    %put WHERE option;
    DATA var10_1;
      SET all (Where=(extravar>0.5));
    RUN;
  %end;

%mend keeploop;

%keeploop

/* End of Program */
```

Six scenarios were used (B1, B2, B3, B4, B5 and B6) to examine the difference in performance between the use of the WHERE option and the IF statement after 25 runs. A summary of the results is presented in Table 2.

The percentage increase in performance of the WHERE option vs. the IF statement is calculated as:

100(real time IF statement – real time WHERE option)/(real time IF statement) (%)*

Scenario B1: A simulated dataset with 10 million observations with approximately 99% of the observations to be selected.

Scenario B2: A simulated dataset with 10 million observations with approximately 75% of the observations to be selected.

Scenario B3: A simulated dataset with 10 million observations and approximately 50% of the observations of the records selected.

Scenario B4: A simulated dataset with 10 million observations with approximately 25% of the observations to be selected.

Scenario B5: A simulated dataset with 10 million observations with approximately 1% of the observations of the records selected.

Scenario B6: A simulated dataset with 10 million observations with approximately 0.1% of the observations to be selected.

Table 2 below shows an overview of the difference in performance (%) after 25 runs for each of the six scenarios.

Table 2: Performance gain of using WHERE option instead of IF statement after 25 runs.

Scenario	$N_{selected}$: $N_{observations}$	Performance gain	
		Mean (%)	SD (%)
B1	0.99	-2.0%	2.7%
B2	0.75	-1.2%	2.3%
B3	0.50	-1.2%	2.3%
B4	0.25	3.0%	1.6%
B5	0.01	7.6%	4.4%
B6	0.001	15.2%	2.7%

SD = Standard Deviation, $N_{selected}$ is the number of selected observations and $N_{observations}$ is the number of observations in the input dataset.

In conclusion, after the six scenarios, it can be said that the larger the percentage of the selected observations is the smaller the gain in performance gets when comparing the WHERE option with the IF statement. With an additional note that there is no performance gain anymore if about more than 50% of the observations is selected.

CONCLUSIONS

- Although there is a difference in performance between the KEEP option (on the SET statement) and the KEEP statement in SAS data steps in favor of the KEEP option, the difference may not be as large as one may expect. However, when a smaller fraction of input variables needs to be kept, the difference in performance is increasing, in favor of the KEEP option (on the SET statement).
  In general, the smaller the ratio N_{keep}: N_{var} is, the more significant the gain in performance of the KEEP option as compared to the KEEP statement becomes.

- There is a difference in performance between the IF statement and the WHERE option (on the SET statement) in a SAS data step in favor of the WHERE option. But, in this case there is even no difference in performance if approximately more than half of the observations is selected. However, if a real small amount of observations is selected then there is a more significant gain in performance, although, again, not as large as one may expect.
  Here it can be said that the ratio $N_{selected}$: $N_{observations}$ determines the performance gain. In general, the smaller this ratio is, the higher the performance gain becomes in favor of the WHERE option (on the SET statement).

NOTE: All performances were measured using SAS® Life Science Analytics Framework (LSAF) 5.2 SAS 9.4M6 / 5.3 SAS 9.4M7.

REFERENCES:

[1] SAS Global Forum 2012, paper 357-2012: Top Ten SAS® Performance Tuning Techniques, Kirk Paul Lafler, Software Intelligence Corporation, Spring Valley, California.

https://support.sas.com/resources/papers/proceedings12/357-2012.pdf

CONTACT DETAILS

Your comments and questions are valued and encouraged. Contact the author at:

Author Name Ruurd Bennink
Company OCS Life Sciences
Address Ruwekampweg 2G
City / Postcode 's-Hertogenbosch / 5222 AT
Work Phone 0031 (0)73 523 6000
Email sasquestions@ocs-consulting.com
Web www.ocs-lifesciences.com