

TS-562

PROC FASTCLUS may give slightly different results on different operating systems

Machines that have floating point registers with more precision than regular floating point storage and that round when transferring values from register to memory may in some rare cases get slightly different final clustering results than machines that truncate or that have floating point registers of the same precision as regular memory. This difference might be found on MVS versus Windows, for example. It will be up to the researcher to decide whether or not the difference, if it occurs, has any practical significance or not.

If you override the default PROC FASTCLUS seed selection process (specifying your own cluster seeds with the SEED= option), and if you happen to specify seeds such that two or more clusters remain empty, the clustering results on some machines (Windows, OS/2, Unix with a pc chip) may be slightly different than the results on other operating systems with a different floating-point register (MVS, Unix, etc).

The different clustering results may occur as a result of a difference in the way the floating-point processing units deal with assigning observations to clusters with identical seeds during the iteration process. If you have multiple seeds that are identical during a cluster assignment phase, call them say A, B, and C, MVS will assign all of the observations that should be assigned to any one of them to ONLY one of them, A say, leaving B and C empty. A PC may instead assign some of the observations to each of the three seeds.

There is no widely held convention as to which method is the statistically preferred method for dealing with the ties, so there is no issue of one solution being correct and the other solution being incorrect.

The only ways to tell if this assignment has changed the final results would be to either examine the cluster assignments at the end of each iteration, or to run the analysis on both types of machines (for example a pc and non-pc operating system) to compare the results.

If you wanted to make sure that the pc results match non-pc results, you would have to examine the cluster seeds at the end of each iteration to make sure that all of the observations going to multiple duplicate seeds get assigned (perhaps manually) to only one seed.

Here is an illustration.

```

data raw;
  do i=1 to 20;
    x=rannor(23232) + 3*mod(i,5);
    output;
  end;
run;
data seeds;
  do x=0,3,6,-30,30;
    output;
  end;
run;
%macro stopit(stop);
  proc fastclus data=raw seed=seeds maxc=5 strict=3 converge=0 maxiter=&stop
    mean=iterstop noprint;
    var x;
  run;
  data iterstop;
    _run_="maxiter=&stop";
    set iterstop;
    drop cluster _radius_ _near_ _gap_;
  run;
  proc datasets nolist;
    append data=iterstop base=all;
  run;
%mend stopit;
%stopit(0)
%stopit(1)
data all;
  if _n_ eq 1 then put _page_
    "          crit  freq  rmsstd      x" /;
  set all;
  by _run_;
  put _run_ _crit_ 7.3 _freq_ 5. _rmsstd_ 9.4 x 8.3;
  if last._run_ then put;
run;

```

SAS Release 6.12 TS045 running on Windows NT produced the following output.

	crit	freq	rmsstd	x
maxiter=0	1.427	4	0.5795	0.152
maxiter=0	1.427	3	0.1670	1.804
maxiter=0	1.427	8	1.5385	7.031
maxiter=0	1.427	0	.	.
maxiter=0	1.427	0	.	.
maxiter=1	1.063	4	0.5795	0.152
maxiter=1	1.063	3	0.1670	1.804
maxiter=1	1.063	7	1.2818	7.841
maxiter=1	1.063	1	.	5.525
maxiter=1	1.063	1	.	4.968

Note that at the end of the maxiter=0 step, there are two clusters with no observations assigned to them (freq=0). PROC FASTCLUS will assign the overall mean of the variables as seed for these clusters – i.e. they will have identical seeds. At the end of the maxiter=1 step, one observation was assigned to cluster 4, and one observation was assigned to cluster 5.

SAS Release 6.12 TS020 running on HP Unix produced the following output.

	crit	freq	rmsstd	x
maxiter=0	1.427	4	0.5795	0.152
maxiter=0	1.427	3	0.1670	1.804
maxiter=0	1.427	8	1.5385	7.031
maxiter=0	1.427	0	.	.
maxiter=0	1.427	0	.	.
maxiter=1	1.063	4	0.5795	0.152
maxiter=1	1.063	3	0.1670	1.804
maxiter=1	1.063	7	1.2818	7.841
maxiter=1	1.063	2	0.3936	5.246
maxiter=1	1.063	0	.	.

At the end of the maxiter=0 step, the results are identical to the Windows NT run, and PROC FASTCLUS will assign seeds as it did before. The difference occurs at the end of the maxiter=1 step – note that two (all) observations were assigned to cluster 4 and no observations were assigned to cluster 5. This is where the difference occurs.

From this point on, the final clustering results may or may not match at the end, and that all depends on the data. With this particular set of data, the results did indeed match of their own accord, and that will usually be the case. To force the results to be the same on the two different machines, find the maxiter=k step in which the results differ, pick the solution of your choice, and use those means as seeds for a new PROC FASTCLUS run.