STAT 541

Chapter 23: Selecting Efficient Sorting Strategies

Outline

- Avoiding Unnecessary Sorts
- Using a Threaded Sort
- Calculating and Allocating Sort Resources (not covered)
- Handling Large Data Sets
- Removing Duplicate Observations Efficiently

Avoiding Unnecessary Sorts

- Sorts can be avoided in some situations
- BY groups with an Index
  - If a data set includes an index, you can use a BY statement on the indexed variable without having used PROC SORT
  - The BY statement can be used in a DATA step or PROC step
  - Processing a data set with an index may be less efficient than PROC SORT
  - Does not index if DESCENDING or NOTSORTED are used or data is pre-sorted

Avoiding Unnecessary Sorts

- The NOTSORTED option groups the data on the BY variable, but doesn't order groups
  - Useful when sorting on nominal groupings
  - Results are interesting when data is not pre-grouped

```
proc freq data=stat541.fall2008;
by gender notsorted;
table major;
run;
```

Avoiding Unnecessary Sorts

- You can actually group on formatted values rather than the variable itself
- GROUPFORMAT option can only be used in the DATA step
- GROUPFORMAT allows you to create groups without creating a new variable
- The CLASS statement is an under-used resource, especially in PROC MEANS and PROC UNIVARIATE

Avoiding Unnecessary Sorts

- PROC CONTENTS can be used to see whether data is already sorted
- The SORTED BY option can then be used to include the sort information as a data set attribute

Using a Threaded Sort

- Threaded sorts can distribute sorting across multiple CPUs

PROC SORT dsname THREADS|NOTHREADS;

- You can modify or query the number of CPUs with CPUCOUNT

Handling Large Data Sets

- If a data set is too large to sort (insufficient space for the multiple copies of the data set needed for a sort), the data set can be split into smaller data sets then reassembled, typically with a SET statement/BY statement combination.

Handling Large Data Sets

- Many methods are available
  - FIRSTOBS= OBS= in DATA step
  - IF/OUTPUT in DATA step
  - WHERE in PROC SORT step
  - WHERE in DATA step
- A DATA step is better than PROC APPEND for reassembling a large data set

Handling Large Data Sets

- The TAGSORT option saves only the BY variables and observation numbers in *temporary* files
- This saves on the space set aside for a SORT

Removing Duplicate Observations Efficiently

- NODUPKEY
- NODUPRECS
  - Checks the entire record, not just the BY variables
  - Can be limited to post-DROP and post-KEEP variables
- FIRST. and LAST.
  - I'm surprised the book included this choice