STAT 541

Chapter 21: Controlling Data Storage Space

Outline

- Reducing Data Storage Space
- Compressing Data Files
- Using Views to Conserve Data Storage

Reducing Data Storage Space

- Character variables
  - Reminder: The LENGTH statement should be listed immediately after the DATA step (even before a SET command) to take effect
  - Use codes rather than lengthy character variables where possible

Reducing Data Storage Space

- Numeric variables
  - The LENGTH variable should only be used with integers, since it otherwise truncates significant digits from the numeric variable (sign, exponent, mantissa)
  - DEFAULT= assigns a default length to *all* subsequent numeric variables, and hence should be used with caution
  - PROC COMPARE can summarize rounding errors

Compressing Data Files

- Uncompressed Data files have several inefficiencies
  - Column space is constant for each record
  - Observation lengths are equal
  - Character variables are padded with blanks
  - Numeric variables are padded with 0s in the mantissa
  - New observations may cause an entire new page to be created

Compressing Data Files

- Compressed Data files have efficiencies that you might anticipate, as well as some that would surprise you
  - Observations are treated as a string of bytes
  - Blanks are removed
  - Consecutive repeated characters and numbers are compressed
  - Information on updated observations is not necessarily stored on the same page
- Greater overhead is required (e.g., pointers)

Compressing Data Files

- Rules for when to compress data sets are intuitive:
  - Large data sets
  - Many long character variables
  - Many repeated character/numeric variables
  - Many missing values
  - Many consecutive repeated character/numeric variables

Compressing Data Files

- Two options (and accompanying suboptions) for compressing files
- OPTIONS COMPRESS=NO|YES|CHAR|BINARY
  - System compress (affects *every* data set in your SAS session)
- DATA dsname (COMPRESS=NO|YES|CHAR|BINARY)
  - Data set compress
  - YES and CHAR are good for simple character repeats
  - BINARY is efficient for long observations, and data with large blocks of numeric variables (e.g., testing data)
  - BINARY requires more CPU to uncompress
- SAS writes a message to LOG summarizing compression

Compressing Data Files

- By default, new observations are appended to the end of a data set (implicit OUTPUT). REUSE allows SAS to repurpose accumulated empty space in the compressed data set
- OPTIONS REUSE=NO|YES
  - System reuse
- DATA dsname (COMPRESS=YES REUSE=YES|NO)
  - Data set reuse
- Once selected, the REUSE option cannot be changed

Compressing Data Files

- Remember the use of POINT= when creating random samples in Chapter 13? This *direct access* has high overhead for compressed data sets and can be disabled with POINTOBS=NO to prevent this inefficient access technique

DATA step views

- We introduced SQL views (partially compiled tables) in Chapter 7 as an important space-saving measure
- Views can be created in the DATA step as well (and are distinct from SQL views):

```
DATA dsname/VIEW=dsname;
DATA VIEW=dsname; DESCRIBE;
```

DATA step views

- Remember that views are not a panacea—they should not be called multiple times in a program since they have to read anew their source data each time the view is referenced.
- Consider saving the view in another data set instead, then referencing that data set instead of the view
- Views must be kept current as underlying data sets change, which creates additional overhead.
- Don't create views that use files whose variable names/length/labels often change