Using an index for efficient WHERE processing
Identifying available indexes
Identifying conditions that can be optimized
Estimating the number of observations
Comparing probable resource usage
Deciding whether to create an index
Comparing procedures that produce detail reports
Comparing tools for summarizing data
Using an Index for Efficient WHERE Processing

- A WHERE statement can use sequential access or direct access (e.g., with an index) to search observations.
- An index is effective when the WHERE group is small.
- There is overhead associated with indexes.
Identifying Available Indexes

- SAS will use an index for a variable in a WHERE statement only if
  - The variable is the key variable in a simple index
  - The variable(s) is(are) the first variable(s) in a composite index
- SAS will use the same index for WHERE and BY statements when they are both present
- Consecutive ordering in a composite index is important
Identifying Conditions that can be Optimized

- WHERE conditions will not be tested for optimization with an index if they contain:
  - functions other than TRIM or SUBSTR
  - SUBSTR, under certain conditions
  - =* (sounds like)
  - arithmetic operators
  - variable-to-variable comparisons

- Compound WHERE conditions have additional constraints
Estimating the Number of Observations

- SAS estimates the subset size specified by the WHERE condition in deciding to use an index.

<table>
<thead>
<tr>
<th>Percentage of Data Set</th>
<th>SAS Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-3%</td>
<td>Direct Access</td>
</tr>
<tr>
<td>3-33%</td>
<td>Probably Direct Access</td>
</tr>
<tr>
<td>33%-100%</td>
<td>Probably Sequential Access</td>
</tr>
</tbody>
</table>

- SAS actually stores quantiles with indexes to help estimate subset size.
Comparing Probable Resource Usage

- Direct access will always be more costly in retrieving data
- SAS compares the number of predicted I/O swaps for direct access vs the number of I/O swaps for sequential access to decide whether to use an index
- Other factors can affect I/O swaps (e.g., order of the data, whether data is compressed)
Deciding Whether to Create an Index

- Do not create an index when the file is small
- Indexes do require overhead—do not create them needlessly
- Sort the data by the index variables before using the index