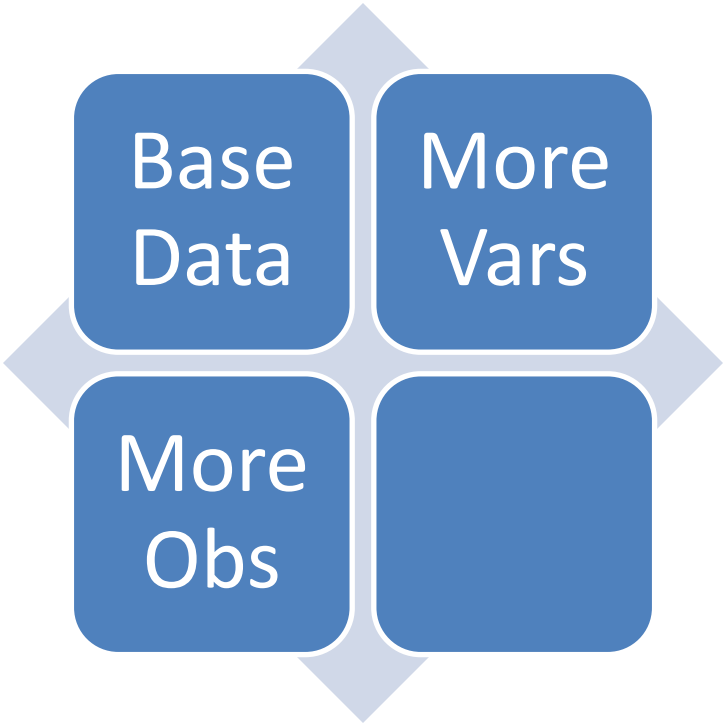
SAS 7 – Combining Data Sets



Basic Merges

- Add observations (“append”, “concatenate”)
- Add variables (“merge”)

Documentation

- Base SAS
  - Base SAS 9.2 Procedures Guide
  - Concepts
  - Reading, Combining, and Modifying SAS Data Sets
- Base SAS
  - SAS 9.2 Language Reference: Dictionary
  - Dictionary of Language Elements
  - Statements

Add observations

- Basic

```
libname y "y:\sas\data";

data combined;
   set y.animal y.plant;
   run;
```

Details of Add Obs

- Can “set” *many* data sets
- Aligns variables of the same name and type
  - If names don’t match, they become separate variables (use “rename” dataset option)
  - If types don’t match, you get an ERROR
  - If other attributes (labels, formats) don’t match, the first one wins
- Variables in one data set but not the other are filled out with missing values

Large Data Sets

- DATA … SET individually processes each observation in every data set.
  - PROC APPEND gains speed by only processing observations from the additional data sets

- If the end goal is a *sorted* data set, interleaving may process faster

```
data interleaved;
   set y.animal y.plant;
   by common;
   run;
```

Add variables

- Two methods of matching observations
  - By position/observation number (avoid this if possible)
  - By key variable(s)
    - One-to-one
    - Many-to-one or one-to-many (a.k.a. table lookup)

Merge by position

- Basic

```
data new;
   merge y.animal y.plant;
   run;

proc print; run;
```

Details

- If there are variables common to both data sets, the data from the last data set wins (including missing values)
- All the data is processed, even if there are a different number of observations in both data sets. It doesn't matter which data set is longer.
- Variables appear in the order of their originating data sets

Merge by key

- Basic

```
data new;
   merge y.animal y.plant;
   by id;
   run;

proc print; run;
```

Details

- Prerequisites
  - Key variable with the same name (and type)
  - Both input data sets must be sorted

- All the data is used
  - If a key value appears in one data set but not the other, it is simply left unmatched. It doesn't matter which input data set is unmatched.
  - (Keep track of which input data set contributes to an observation with the "in" data set option.)

More details

- If key values are unique within each data set, this produces a one-to-one match
- If key values are unique within one data set but repeated in the other, this produces one-to-many or many-to-one matches (data set order doesn't matter)
- Many-to-many matches are also possible (but unusual … be careful not to do this accidentally)
- Keys may be composed of more than one variable ("compound" keys)