PSTAT 130

SAS BASE PROGRAMMING

- Lecture 7 -

Dr. Julie Swenson

Objectives

- Combine Data Sets
  - Concatenate Data Sets (Appending)
    - The SET statement
  - Merge Data Sets (Merging)
    - The MERGE and BY statements
    - Types of Merging

Why Separate Data Sets?

- Efficiency of Storage
  - Keep only the data you need in each data set
    - Example: Customer information is separate from order information
- Efficiency of Processing
  - Smaller data sets can be processed faster than larger data sets
    - Example: Reading in fewer variables or sorting fewer observations

Combine SAS Data Sets

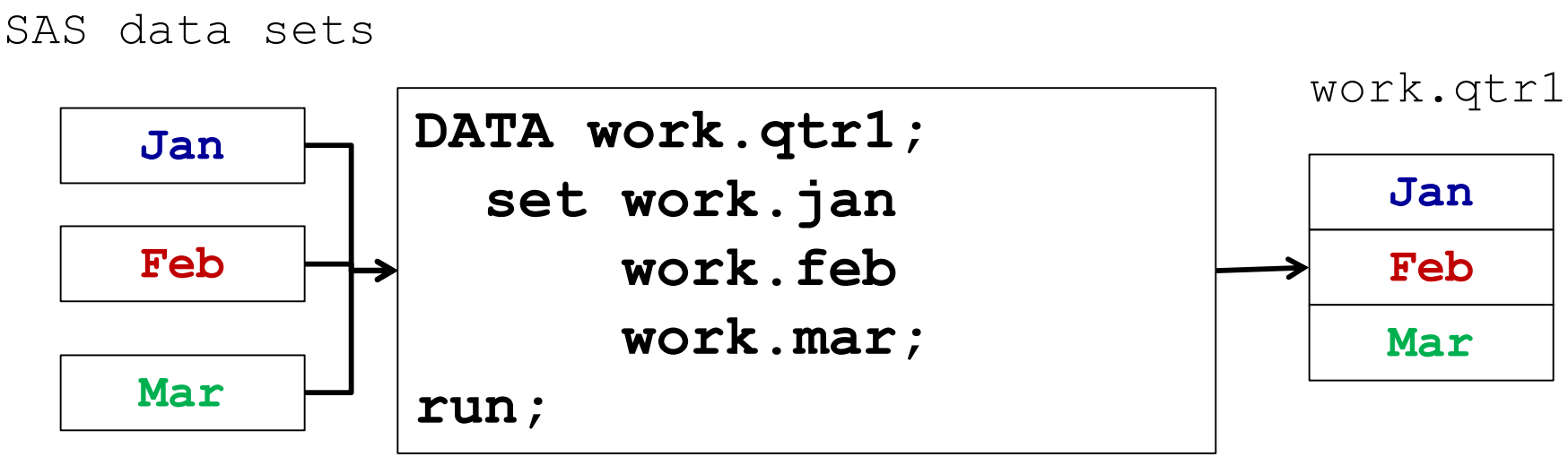
- Append (or concatenate): SET statement
  - Two data sets each with the SAME variables
    - Data set A has m observations and k variables
    - Data set B has n observations and k variables
    - Combined data set has $(m+n)$ observations and k variables

Concatenate SAS Data Sets

- General form of concatenating data sets

```
DATA Output-SAS-data-set;
   SET SAS-data-set1 SAS-data-set2 . . .;
   <other SAS statements>
RUN;
```

Concatenate SAS Data Sets



Concatenate Data Sets: Example

- I have two sections of my class, Morning and Afternoon. I want to combine their scores as follows:

Morning

Name	Score
Mary	75
Mark	82
Mike	68

allsections

Name	Score
Mary	75
Mark	82
Mike	68
Andy	78
Alice	85
Art	62

Afternoon

Name	Score
Andy	78
Alice	85
Art	62

Concatenate Data Sets: Example

- I have two sections of my class, Morning and Afternoon, and want to combine their scores.

Morning

Name	Score
Mary	75
Mark	82
Mike	68

Afternoon

Name	Score
Andy	78
Alice	85
Art	62

- What happens if we run the following step?

```
data allsections;
   set afternoon morning;
run;
```

Concatenate Data Sets: Example

- I have two sections of my class, Morning and Afternoon, and want to combine their scores.

Morning

Name	Score
Mary	75
Mark	82
Mike	68

Afternoon

Name	Score
Andy	78
Alice	85
Art	62

- What happens if we run the following step?

```
data allsections;
   set morning afternoon;
run;
```

Concatenate Data Sets: Overview

- To append (or concatenate) data sets
  - The variable names and data types should be the same in both data sets.
  - You can append multiple data sets in a single SET statement.

- Note: You may want to create a variable that identifies the source of each observation. Do this in a separate data step *prior* to appending the data sets.

Create an Identifier Variable

```
data morning;
   input name $ score;
   class = 'M';
datalines;
…
run;

data afternoon;
   input name $ score;
   class = 'A';
datalines;
…
run;

data allsections;
   set morning afternoon;
run;
```

Name	Score	Class
Barry	82	M
Mary	75	M
Zach	68	M
Alice	85	A
Art	62	A
Will	78	A

Concatenate Data Sets

- What if the variables in the data sets (to be appended) have different attributes? (i.e. labels, formats)

Example

```
data morning;
input name $ birthdate mmddyy10.;
format birthdate mmddyy8.;
datalines;
Mark 01/12/1981
Mike 02/15/1983
Mary 03/19/1982
;
run;

data afternoon;
input name $ birthdate mmddyy10.;
format birthdate date9.;
datalines;
Abby 01/12/1981
Alice 02/15/1983
Art 03/19/1982
;
run;

data combined1;
  set morning afternoon;
run;
proc print data=combined1;
run;

data combined2;
  set afternoon morning;
run;
proc print data=combined2;
run;
```

Combine SAS Data Sets

- Match Merge: MERGE & BY statements
  - Two data sets with at least one common variable and other unique variables
    - Data set A has *m* observations and *k* UNIQUE variables
    - Data set B has *n* observations and *j* UNIQUE variables
    - Combined data set has <u>at most</u> *m+n* observations (typically much less) and *k+j+1* variables (in the case of <u>one</u> common variable)

Combine Data Sets

- Match Merge: MERGE & BY statements
  - The observations from each data set with the same value of the (unique) BY variable are linked and output as one observation
  - If you omit the BY statement, the first observation from each data set are output together as one observation without being linked by a common variable

Match Merge Data Sets

- General form of a DATA step match-merge

```
DATA SAS-data-set;
   MERGE SAS-data-sets;
   BY BY-variable(s);
   <other SAS statements>
RUN;
```

- Note: Data sets must be sorted on BY variable prior to merging.

Merge Data Sets: Example

- I have midterm scores in one data set and final scores in another. I want to combine them as follows:

Midterm

Name	Midscore
Wendy	32
Andy	38
John	27

allscores

Name	Midscore	Finalscore
Andy	38	82
John	27	91
Wendy	32	73

Final

Name	Finalscore
John	91
Wendy	73
Andy	82

Merge Data Sets: Example

- I have midterm scores in one data set and final scores in another, and want to combine them.

Midterm

Name	Midscore
Wendy	32
Andy	38
John	27

Final

Name	Finalscore
John	91
Wendy	73
Andy	82

- What happens if we run the following step?

```
data allscores;
   merge midterm final;
run;
```

Merge Data Sets: Example

- I have midterm scores in one data set and final scores in another, and want to combine them.

Midterm

Name	Midscore
Wendy	32
Andy	38
John	27

Final

Name	Finalscore
John	91
Wendy	73
Andy	82

- What happens if we run the following step?

```
data allscores;
   merge midterm final;
   by name;
run;
```

Match Merge Data Sets: Overview

- To match merge data sets
  - At least one common variable must exist in all the data sets.
  - Some unique variables should exist in the data sets.
  - You can merge multiple data sets in a single MERGE statement.
  - You will need to pre-sort the data by the desired BY variable.

Types of Merges

- Match Merge: MERGE & BY statements
  - One-to-one
    - Unique BY values in one data set and *unique* matching BY values in the other data set
  - One-to-many
    - Unique BY values in one data set and *duplicate* matching BY values in the other data set
  - Many-to-many
    - Duplicate matching BY values in both data sets

One-to-Many Merge

work.one

X	Y
1	A
2	B
3	C

work.two

X	Z
1	A1
1	A2
2	B1
3	C1
3	C2

```
data work.three;
    merge work.one work.two;
    by X;
run;
```

work.three

X	Y	Z
1	A	A1
1	A	A2
2	B	B1
3	C	C1
3	C	C2

One-to-Many Merge

ia.allsales

Month	Region	Sales
1	Europe	2118222
1	North America	3135765
2	Europe	1960034
2	North America	2926929

ia.allgoals

Month	Goal
1	2127742
2	1920751
3	2125112

```
data ia.allcompare;
   merge ia.allsales ia.allgoals;
   by Month;
   Difference=Sales-Goal;
run;
```

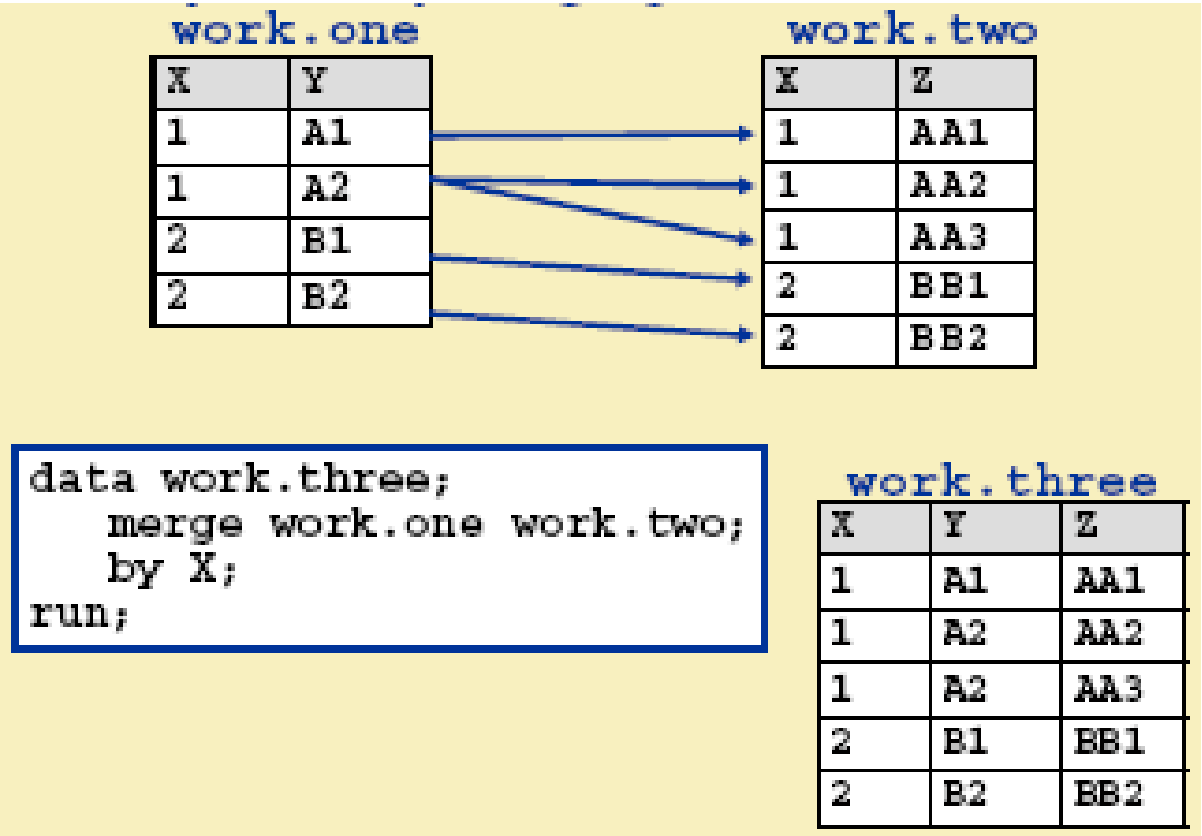
ia.allcompare

Month	Region	Sales	Goal	Difference
1	Europe	2118222	2127742	-9520
1	North America	3135765	2127742	1008023
2	Europe	1960034	1920751	39283
2	North America	2926929	1920751	1006178

Example: Parent-Child Tables

- Sales Order Form
  - Parent Table contains Order-level information
    - Order #, Order Date, Name, Phone Number, Shipping Address, Billing Address, Payment Method, etc.
  - Child Table contains
    - Order #, List of items ordered, Quantity, Unit cost
  - Order # appears on every observation in Parent and Child tables
  - Tables are linked through One-to-Many merging using the common variable Order #

Many-to-Many Merge



Rename= Data Set Option

- General form of the RENAME= data set option

```
SAS-data-set(RENAME=(old-name-1=new-name-1
                     old-name-2=new-name-2
                                  .
                                  .
                                  .
                     old-name-n=new-name-n))
```

  - When <u>appending</u> data sets, use the RENAME= option to create **common** variable names.
  - When <u>merging</u> data sets, use the RENAME= option to create **unique** variable names.

Appending Example

- Suppose we have, instead of `score` in both data sets, `score` in one data set and `testscore` in the other.

Morning

Name	Score
Mary	75
Mark	82
Mike	68

Afternoon

Name	Testscore
Andy	78
Alice	85
Art	62

- What happens if we run the following step?

```
data allsections;
   set morning afternoon;
run;
```

Rename= Appending Example

- We can try the following:

```
data allsections;
   set morning
       afternoon(RENAME=(testscore=score));
run;
```

Merging Example

- Suppose we have, instead of `Midscore` and `Finalscore`, the variable `score` in both data sets.

Midterm

Name	Score
Wendy	32
Andy	38
John	27

Final

Name	Score
John	91
Wendy	73
Andy	82

- What happens if we run the following step?

```
data allscores;
   merge midterm final;
   by name;
run;
```

Rename= Merging Example

- We can try the following:

```
data allscores;
   merge midterm(RENAME=(score=MidtermScore))
         final(RENAME=(score=FinalScore));
   by name;
run;
```

The IN= Data Set Option

- Use the IN= option to create variables identifying which data sets contain the observation

- General form of the IN= data set option

```
SAS-data-set(IN=variable)
```

- Example

```
data allscores;
   merge midterm(IN=InMidterm)
         final(IN=InFinal);
   by Name;
   if InMidterm and InFinal;
run;
```

Lookup Tables

- Data set variable contains “codes”
  - Males are coded as 1
  - Females are coded as 2

- Lookup table contains “labels” that can be merged with “codes”

GenderLookup

GenderCode	GenderLabel
1	Male
2	Female

Example: Course Scheduling

- The University maintains multiple data sets to schedule classes
  - A list of instructors and the courses they teach.
  - A list of students taking each course.
  - A list of classrooms and the courses that meet in them.

Students

- Variables
  - StudentName
- Three data sets are provided
  - PSTAT130.txt
  - PSYCH118.txt
  - POLI125.txt

```
          1         2
12345678901234567890
John Thomas
Elizabeth Smith
Rajesh Krish
Lily Yang
Robert Williams
Tracy Jones
Cheryl Smith
Alex Shepard
Trinh Phan
Lee Barrett
Clark Johnson
Jenny Page
Mary Marcus
Curt Forrest
Andy Potts
```

Instructors

- Variables
  - InstructorName
  - AcademicRank
  - Salary
  - CourseName
  - FirstClassDate

- File saved as `Instructors.txt`

```
          1         2         3         4         5
12345678901234567890123456789012345678901234567890
John Tukey      Assoc  $56,000  PSTAT130  09/23/10
Sigmund Freud   Assoc  $92,000  PSYCH118  09/24/10
Karl Marx       Asst   $78,000  POLI125   09/27/10
```

Classrooms

- Variables
  - `BldgName`
  - `RoomNumber`
  - `CourseName`
  - `Days`
  - `Time` (read as a character variable)
- File saved as `classrooms.txt`

```
          1         2         3         4
12345678901234567890123456789012345678901234 5
Phelps Hall  222  PSYCH118  T/TH   10:00 am
South Hall   518  PSTAT130  M/W/F   5:00 pm
Phelps Hall  126  POLI125   M/W     2:00 pm
```

Class Exercise

- Create three DATA steps to read in each list of students
  - Create a variable to store the `CourseName` for each data set because only the filename identifies the course name
- Create a DATA step to read in Instructors data
- Create a DATA step to read in the Classrooms data

Class Exercise - continued

- Combine the student lists into a single data set of all students: call it `AllStudents`

- Combine the `Instructor`, `Classrooms`, and `AllStudents` data sets so that each student is paired with his or her instructor and the room information for that class

Class Exercise - continued

- Create a 'Roster' for each class showing the names of the students taking that class.
  - Separate the lists onto a page for each class.
  - Assign appropriate variable labels for `CourseName` and `StudentName`.
  - Show only the `CourseName` at the top of each page and the list of students names below.

```
----------------- Course Name=POLI125 ----------------
                      Student Name

                      Alex Shepard
                      Andy Potts
                      Cheryl Smith
                      Curt Forrest
```

Class Exercise - continued

- Create a 'Class List' for each Student showing details of the classes each is taking.
  - Include the variables below
    - `CourseName`, `FirstClassDate`, `InstructorName`, `BldgName`, `RoomNumber`, `Days`, `Time`
  - Assign appropriate variable labels
  - Use an appropriate format for `FirstClassDate`

Example: Output Class List

```
------------------------------ Student Name=Alex Shepard -------------------------------

                        First
         Course         Class     Instructor     Building       Room     Class     Class
          Name           Date       Name          Name         Number    Days      Time

        PSTAT130     09/23/10    John Tukey    South Hall       518     M/W/F    5:00 pm
        POLI125      09/27/10    Karl Marx     Phelps Hall      126     M/W      2:00 pm


------------------------------ Student Name=Andy Potts -------------------------------

                       First
       Course          Class     Instructor       Building       Room     Class     Class
        Name            Date       Name            Name         Number    Days      Time

      PSTAT130     09/23/10    John Tukey       South Hall       518     M/W/F    5:00 pm
      PSYCH118     09/24/10    Sigmund Freud    Phelps Hall      222     T/TH     10:00 am
      POLI125      09/27/10    Karl Marx        Phelps Hall      126     M/W      2:00 pm
```

Class Exercise - continued

- Create a 'Master List' of all Students whose instructor is an Associate Professor.
  - Include the variables in the example on the next page
  - Assign appropriate variable labels
  - Create a user-defined format and apply it to academic rank, assigning the label “Assistant Professor” to “Asst” and “Associate Professor” to “Assoc”
  - Use an appropriate format for `Salary`

Example: Output

List of Students for Associate Professors

Student Name	Course Name	Instructor Name	Academic Rank	Salary
Alex Shepard	PSTAT130	John Tukey	Associate Professor	$56,000
Andy Potts	PSTAT130	John Tukey	Associate Professor	$56,000
Andy Potts	PSYCH118	Sigmund Freud	Associate Professor	$92,000
Cheryl Smith	PSTAT130	John Tukey	Associate Professor	$56,000
Clark Johnson	PSTAT130	John Tukey	Associate Professor	$56,000
Clark Johnson	PSYCH118	Sigmund Freud	Associate Professor	$92,000
Curt Forrest	PSTAT130	John Tukey	Associate Professor	$56,000
Curt Forrest	PSYCH118	Sigmund Freud	Associate Professor	$92,000
Elizabeth Smith	PSTAT130	John Tukey	Associate Professor	$56,000
Elizabeth Smith	PSYCH118	Sigmund Freud	Associate Professor	$92,000
Jenny Page	PSTAT130	John Tukey	Associate Professor	$56,000
Jenny Page	PSYCH118	Sigmund Freud	Associate Professor	$92,000
John Thomas	PSTAT130	John Tukey	Associate Professor	$56,000
John Thomas	PSYCH118	Sigmund Freud	Associate Professor	$92,000
Lee Barrett	PSTAT130	John Tukey	Associate Professor	$56,000
Lee Barrett	PSYCH118	Sigmund Freud	Associate Professor	$92,000