Combining Data

Shannon Pileggi

STAT 330

OUTLINE

Stacking

Merging

Tracking Observations

Concatenate

Goal: combine multiple data sets that have the same variables

How: use the SET statement to concatenate (or stack) the data sets

Data Set MEN

ID	NAME	GRADE
1	Andrew	B+
3	Jimmy	B-
5	Ulric	A-

Data Set MEN

Data Set WOMEN

ID	NAME	GRADE
2	Soma	B
4	Karen	A
6	Beth	B+

Data Set WOMEN

SAS Code

```
DATA STACKED;
   SET men women;
RUN;
```

SAS Code

Data Set STACKED

ID	NAME	GRADE
1	Andrew	B+
3	Jimmy	B-
5	Ulric	A-
2	Soma	B
4	Karen	A
6	Beth	B+

Interleave

Goal: stack data while retaining some sort of order

How: use the SET statement with BY (data must be pre-sorted)

Data Set MEN

ID	NAME	GRADE
1	Andrew	B+
3	Jimmy	B-
5	Ulric	A-

Data Set MEN

Data Set WOMEN

ID	NAME	GRADE
2	Soma	B
4	Karen	A
6	Beth	B+

Data Set WOMEN

SAS Code

```
DATA INTERLEAVE;
   SET men women;
   BY ID;
RUN;
```

SAS Code

Data Set INTERLEAVE

ID	NAME	GRADE
1	Andrew	B+
2	Soma	B
3	Jimmy	B-
4	Karen	A
5	Ulric	A-
6	Beth	B+

PROC SORT syntax

SAS Code

```
PROC SORT DATA = men ;
   BY ID ;
RUN ;

PROC SORT DATA = women ;
   BY ID ;
RUN ;

DATA DEMO;
SET men women ;
BY ID;
RUN;
```

SAS Code

Original men and women data set are sorted.

SAS Code

```
PROC SORT DATA = men
   OUT = sorted_men ;
   BY ID ;
RUN ;
PROC SORT DATA = women
   OUT = sorted_women ;
   BY ID ;
RUN ;
DATA DEMO;
SET sorted_men sorted_women ;
BY ID;
RUN;
```

SAS Code

Original men and women data sets remain *unsorted*; newly created data sets sorted_men and sorted_women are sorted.

Discussion

Data Set MEN

ID	NAME	GRADE
1	Andrew	B+
3	Jimmy	B-
5	Ulric	A-

Data Set MEN

Data Set WOMEN

ID	NAME	LETTER
2	Soma	B
4	Karen	A
6	Beth	B+

Data Set WOMEN

SAS Code

```
DATA demo;
  SET men women ;
RUN;
```

SAS Code

The goal is for the demo data set to have ____ variables; it would have ____ variables.

1. 2, 3
2. 3, 3
3. 3, 4
4. 4, 3
5. 4, 4

Rename

Goal: stack data with same variables but different names

Data Set MEN

ID	NAME	GRADE
1	Andrew	B+
3	Jimmy	B-
5	Ulric	A-

Data Set MEN

Data Set WOMEN

ID	NAME	LETTER
2	Soma	B
4	Karen	A
6	Beth	B+

Data Set WOMEN

SAS Code

```
DATA DEMO;
SET men
  women (RENAME=(LETTER=GRADE));
BY ID;
RUN;
```

SAS Code

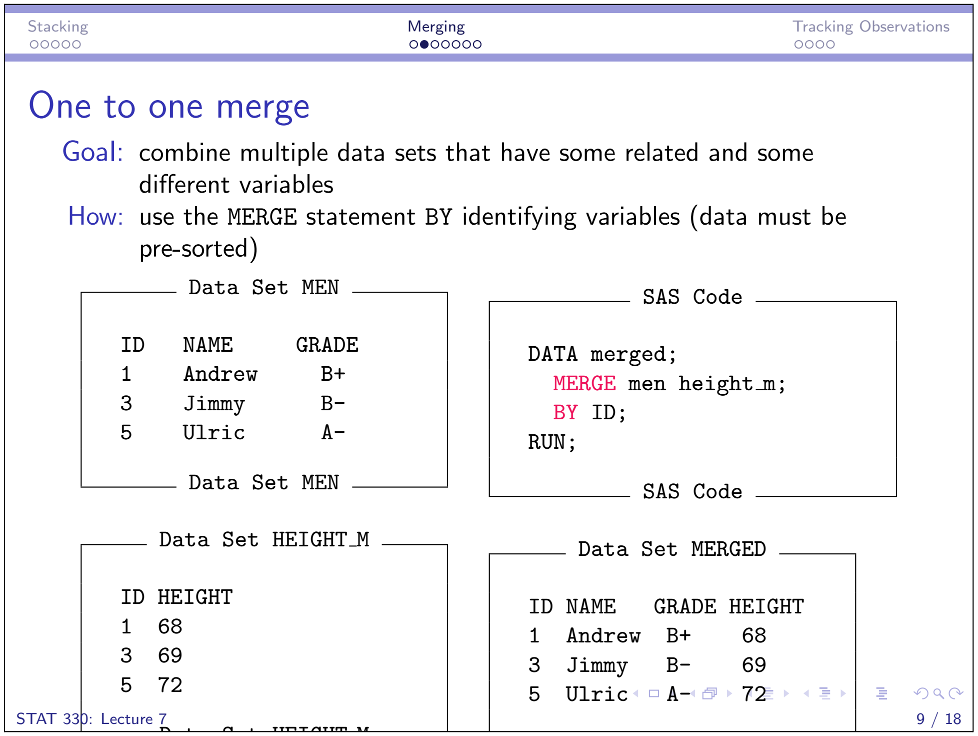
Data Set DEMO

ID	NAME	GRADE
1	Andrew	B+
2	Soma	B
3	Jimmy	B-
4	Karen	A
5	Ulric	A-
6	Beth	B+

Stacking

Merging

Tracking Observations



One to one merge

Goal: combine multiple data sets that have some related and some different variables

How: use the MERGE statement BY identifying variables (data must be pre-sorted)

Data Set MEN

```
ID    NAME      GRADE
1     Andrew      B+
3     Jimmy       B-
5     Ulric       A-
```

Data Set MEN

SAS Code

```
DATA merged;
  MERGE men height_m;
  BY ID;
RUN;
```

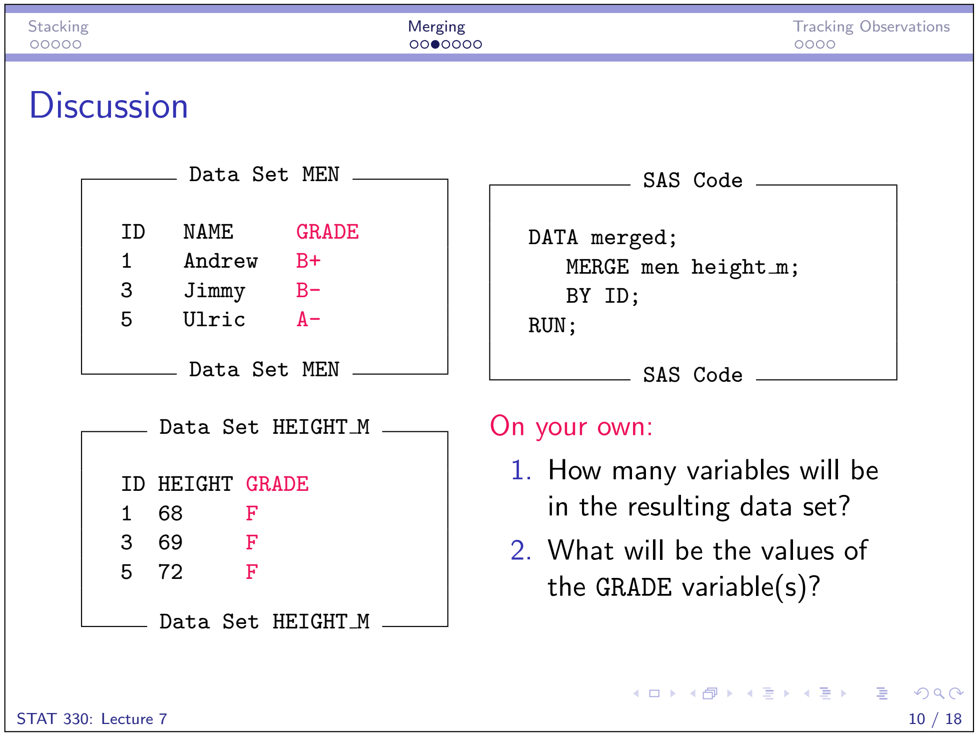
SAS Code

Data Set HEIGHT_M

```
ID HEIGHT
1  68
3  69
5  72
```

Data Set MERGED

```
ID NAME    GRADE HEIGHT
1  Andrew  B+     68
3  Jimmy   B-     69
5  Ulric   A-     72
```



Discussion

Data Set MEN

```
ID   NAME     GRADE
1    Andrew   B+
3    Jimmy    B-
5    Ulric    A-
```

Data Set MEN

SAS Code

```
DATA merged;
   MERGE men height_m;
   BY ID;
RUN;
```

SAS Code

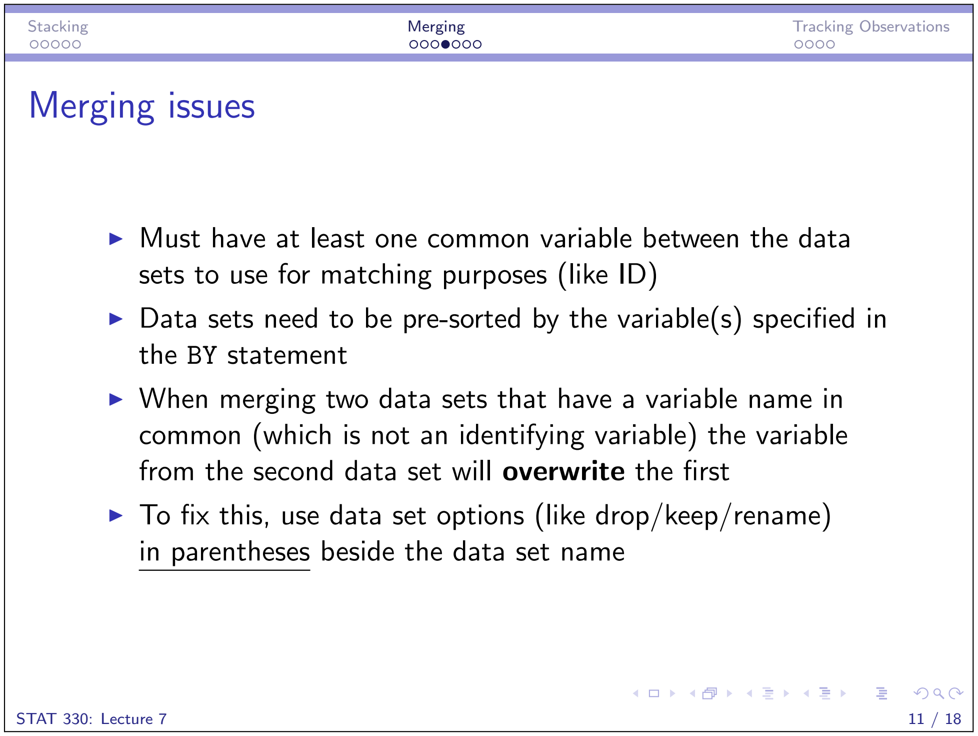
Data Set HEIGHT_M

```
ID HEIGHT GRADE
1  68     F
3  69     F
5  72     F
```

Data Set HEIGHT_M

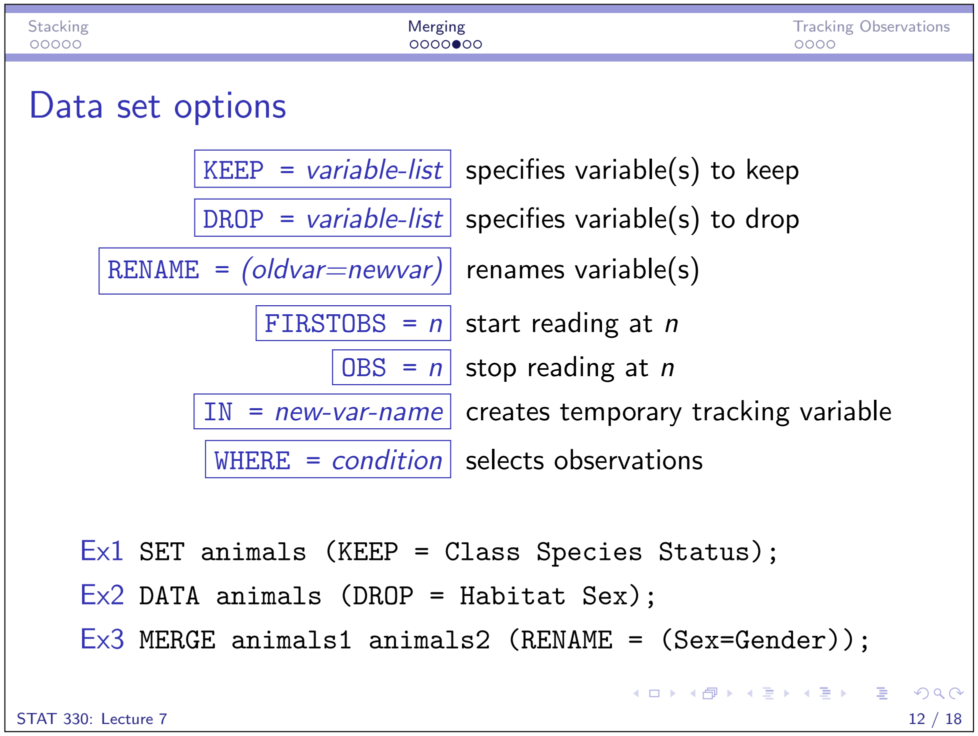
On your own:

1. How many variables will be in the resulting data set?
2. What will be the values of the GRADE variable(s)?



Merging issues

- Must have at least one common variable between the data sets to use for matching purposes (like ID)
- Data sets need to be pre-sorted by the variable(s) specified in the BY statement
- When merging two data sets that have a variable name in common (which is not an identifying variable) the variable from the second data set will **overwrite** the first
- To fix this, use data set options (like drop/keep/rename) in parentheses beside the data set name



Data set options

KEEP = *variable-list*	specifies variable(s) to keep
DROP = *variable-list*	specifies variable(s) to drop
RENAME = *(oldvar=newvar)*	renames variable(s)
FIRSTOBS = *n*	start reading at *n*
OBS = *n*	stop reading at *n*
IN = *new-var-name*	creates temporary tracking variable
WHERE = *condition*	selects observations

Ex1 `SET animals (KEEP = Class Species Status);`

Ex2 `DATA animals (DROP = Habitat Sex);`

Ex3 `MERGE animals1 animals2 (RENAME = (Sex=Gender));`

Discussion

```
 Data Set MEN

 ID   NAME     GRADE
 1    Andrew   B+
 3    Jimmy    B-
 5    Ulric    A-
 8    Allan    B

 Data Set MEN
```

```
 SAS Code

 DATA merged;
    MERGE men height_m;
    BY ID;
 RUN;

 SAS Code
```

```
 Data Set HEIGHT_M

 ID HEIGHT
 1  68
 3  69
 5  72
 10 70

 Data Set HEIGHT_M
```

How many *observations* will be in the resulting data set?

0. none - there will be an error
3. 3
4. 4
5. 5

One to many merge

Goal: combine data sets that have different numbers of observations

How: use the MERGE statement BY identifying variables (data must be pre-sorted)

```
 Data Set PROF

 ID   NAME
 1    Sklar
 3    Doi
 4    Peck

 Data Set PROF
```

```
 SAS Code

 DATA merged;
    MERGE prof class;
    BY id;
 RUN;

 SAS Code
```

```
 Data Set CLASS

 ID CLASS
 1  Stat218
 1  Stat417
 3  Stat150
 3  Stat330
 3  Stat418
 4  Stat251
 4  Stat323
 4  Stat423

 Data Set CLASS
```

```
 Data Set MERGED

 ID  NAME   CLASS
 1   Sklar  Stat218
 1   Sklar  Stat417
 3   Doi    Stat150
 3   Doi    Stat330
 3   Doi    Stat418
 4   Peck   Stat251
 4   Peck   Stat323
 4   Peck   Stat423

 Data Set MERGED
```

Stacking

Merging

Tracking Observations

Tracking with IN=

- When combining data sets, we can track if an individual observation is present/absent in only one data set or in both
- The IN=*new-var-name* option creates a *temporary* indicator variable with values of 0/1

  0 = observation not found in that data set

  1 = observation found in that data set
- Can be used with the SET or MERGE statements, but typically it is used with MERGE
- These indicator variables are typically used for subsetting data
- Visualize this with Venn diagrams:

  http://analisisydecision.es/wp-content/uploads/2014/12/tipos-de-merge-en-SAS.png

Example

```
Data Set MEMBERS

ID  STATE
101 NC
102 CA
103 CA
104 WI
105 NY

Data Set MEMBERS
```

```
Data Set ORDERS

ID  TOTAL
102 30.01
104 254.98
104 75.00
101 1600.56
102 385.30

Data Set ORDERS
```

- iFixit is a local SLO based company
  - provides *free* repair guides (phones, washing machines, etc.)
  - makes money through selling repair tools and parts
- One database stores member information, another stores member orders
- Goal: identify members who haven't made a recent purchase

On your own: What is the data we want?

Discussion

```
SAS Code

DATA example;
   MERGE members (IN=a)
         orders  (IN=b);
    BY id;
    IF condition;
RUN;

SAS Code
```

Which `if` statement should you use keep members who haven't made a recent purchase?

1. `if a;`
2. `if b;`
3. `if a and b;`
4. `if a or b;`
5. `if a and not b;`
6. `if not a and b;`
7. `if not (a and b);`