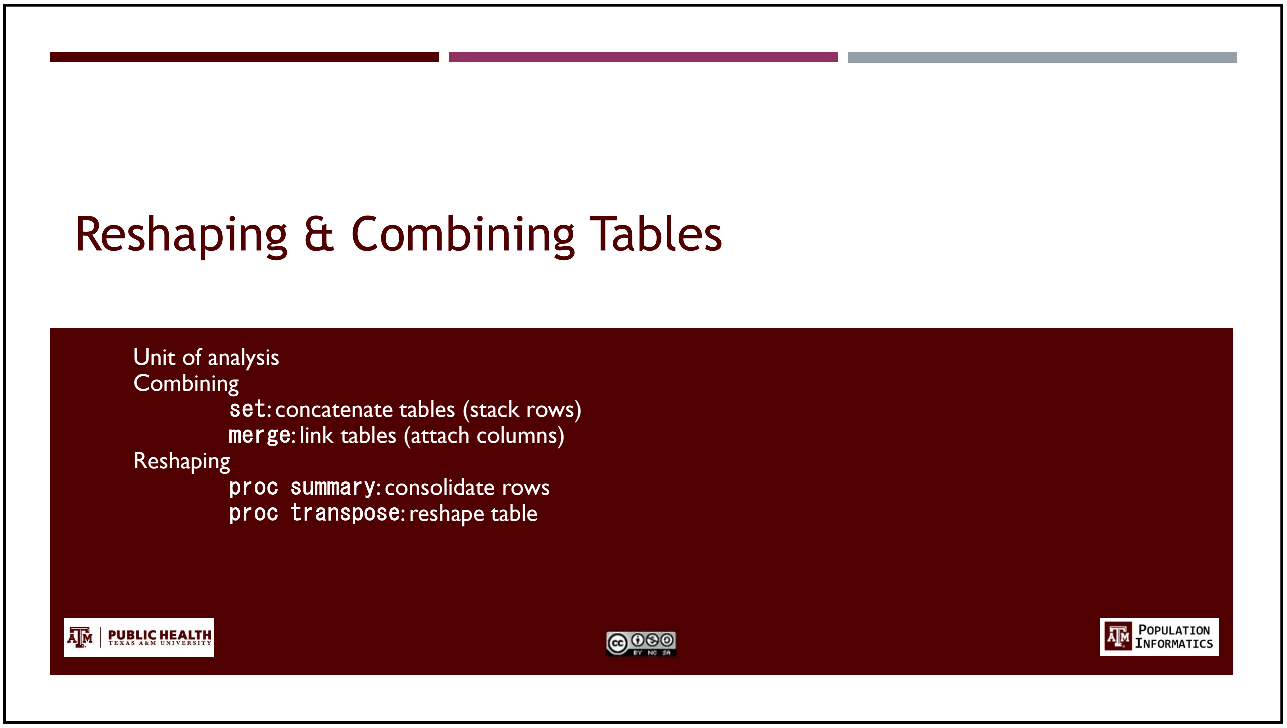


Reshaping & Combining Tables

Unit of analysis

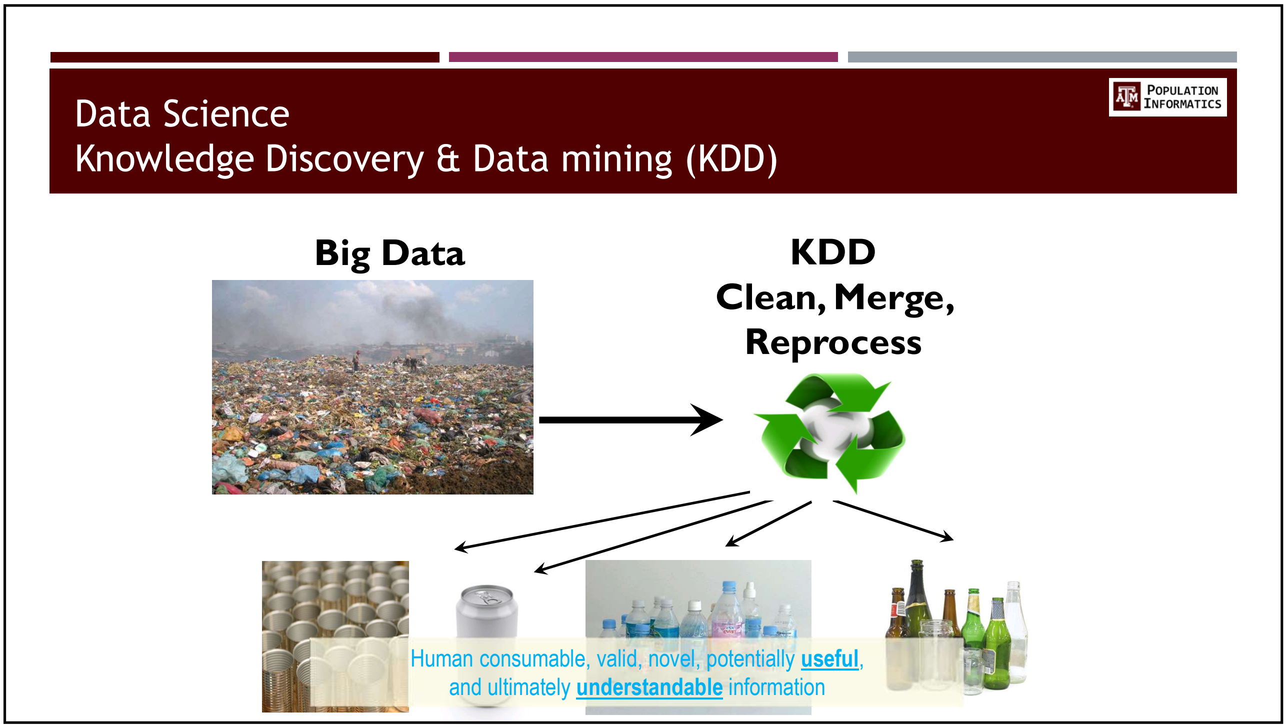
Combining

- `set`: concatenate tables (stack rows)
- `merge`: link tables (attach columns)

Reshaping

- `proc summary`: consolidate rows
- `proc transpose`: reshape table

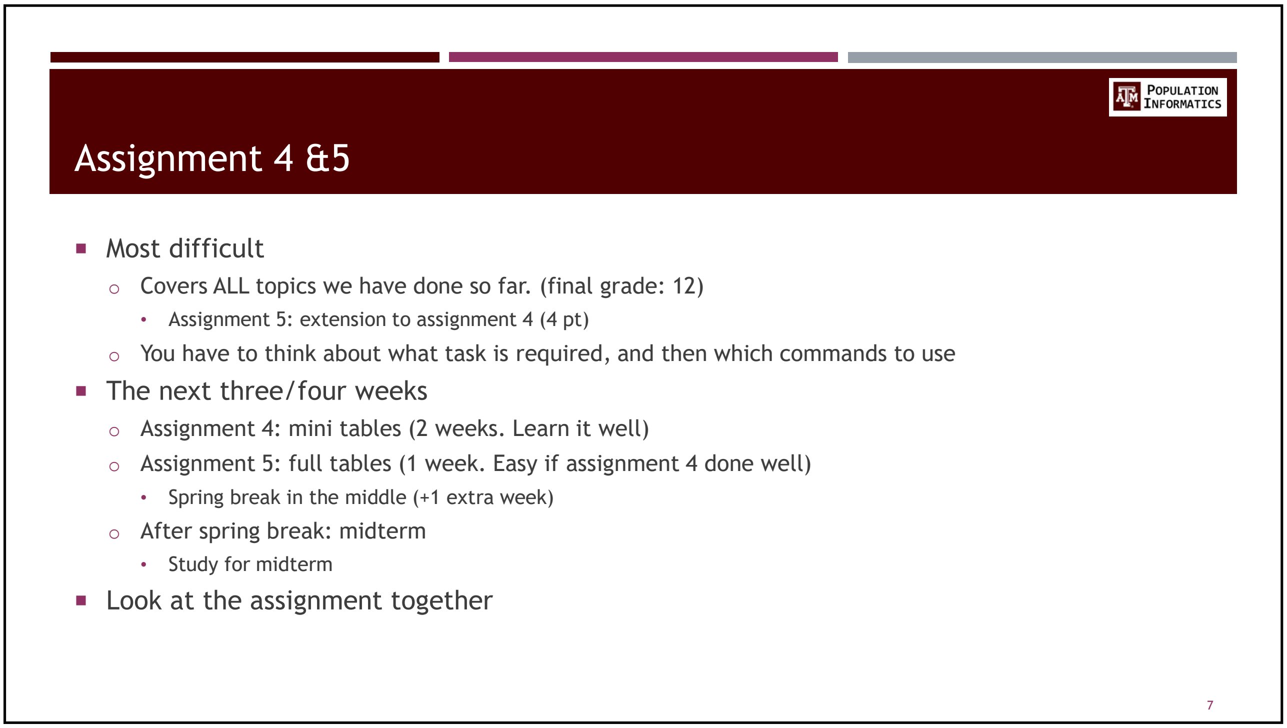
5



Data Science
Knowledge Discovery & Data mining (KDD)

Human consumable, valid, novel, potentially **useful**, and ultimately **understandable** information

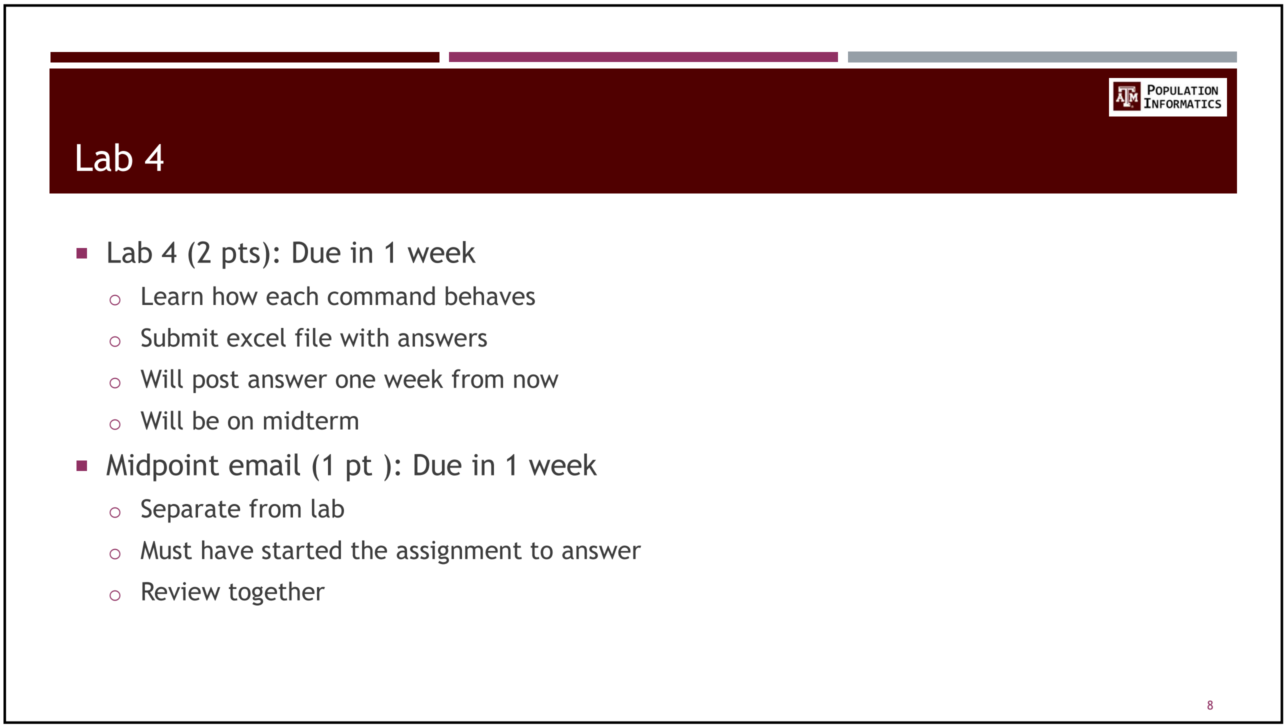
6



Assignment 4 &5

- Most difficult
  - Covers ALL topics we have done so far. (final grade: 12)
    - Assignment 5: extension to assignment 4 (4 pt)
  - You have to think about what task is required, and then which commands to use
- The next three/four weeks
  - Assignment 4: mini tables (2 weeks. Learn it well)
  - Assignment 5: full tables (1 week. Easy if assignment 4 done well)
    - Spring break in the middle (+1 extra week)
  - After spring break: midterm
    - Study for midterm
- Look at the assignment together

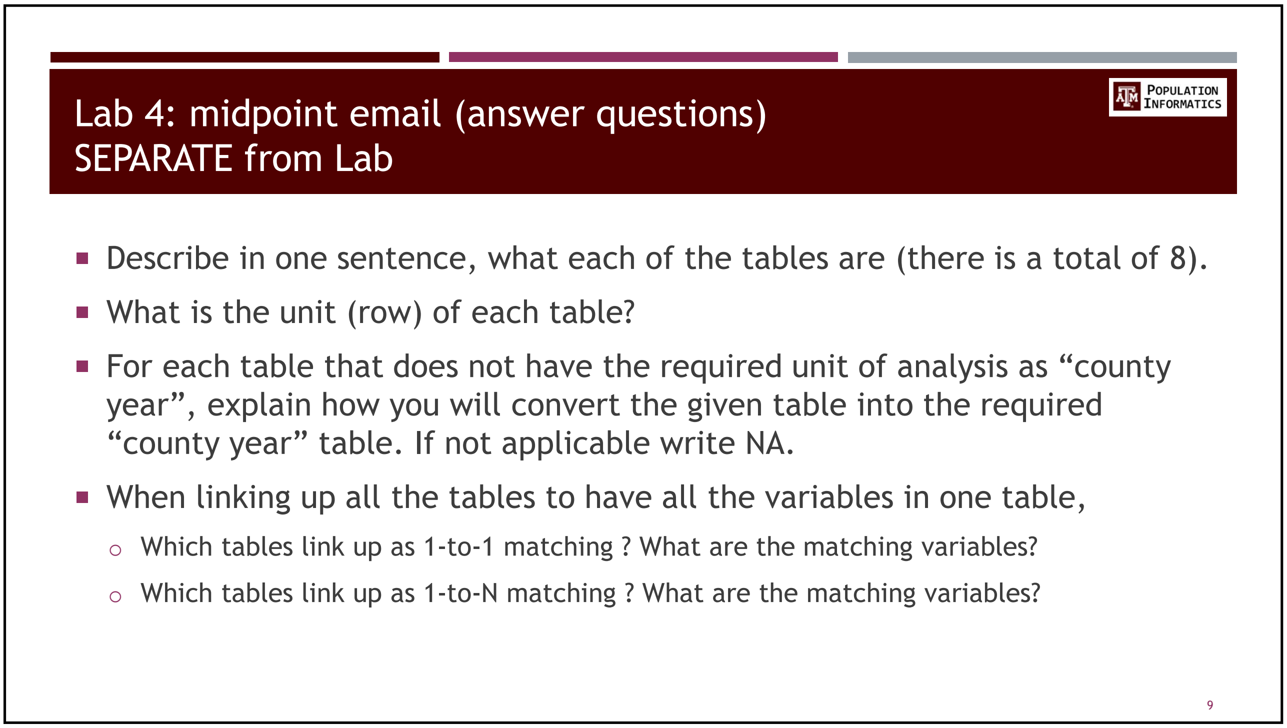
7



Lab 4

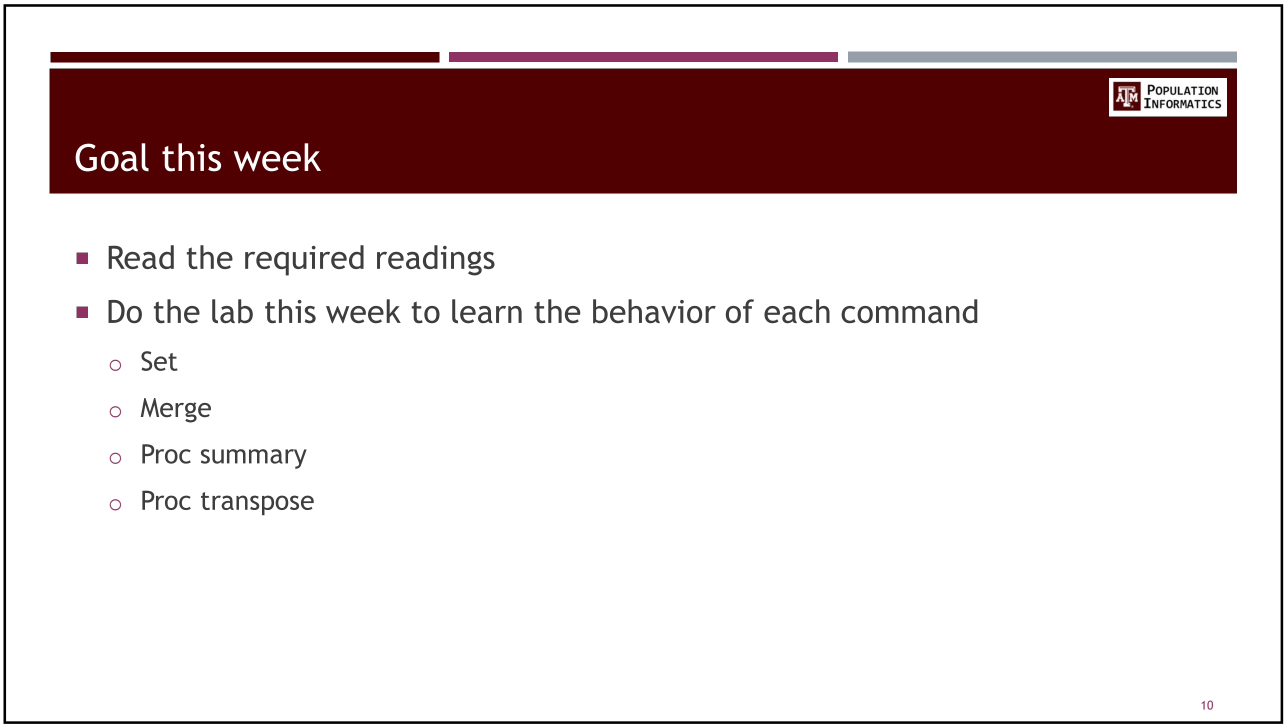
- Lab 4 (2 pts): Due in 1 week
  - Learn how each command behaves
  - Submit excel file with answers
  - Will post answer one week from now
  - Will be on midterm
- Midpoint email (1 pt): Due in 1 week
  - Separate from lab
  - Must have started the assignment to answer
  - Review together

8



Lab 4: midpoint email (answer questions) SEPARATE from Lab

- Describe in one sentence, what each of the tables are (there is a total of 8).
- What is the unit (row) of each table?
- For each table that does not have the required unit of analysis as “county year”, explain how you will convert the given table into the required “county year” table. If not applicable write NA.
- When linking up all the tables to have all the variables in one table,
  - Which tables link up as 1-to-1 matching ? What are the matching variables?
  - Which tables link up as 1-to-N matching ? What are the matching variables?



Goal this week

- Read the required readings
- Do the lab this week to learn the behavior of each command
  - Set
  - Merge
  - Proc summary
  - Proc transpose

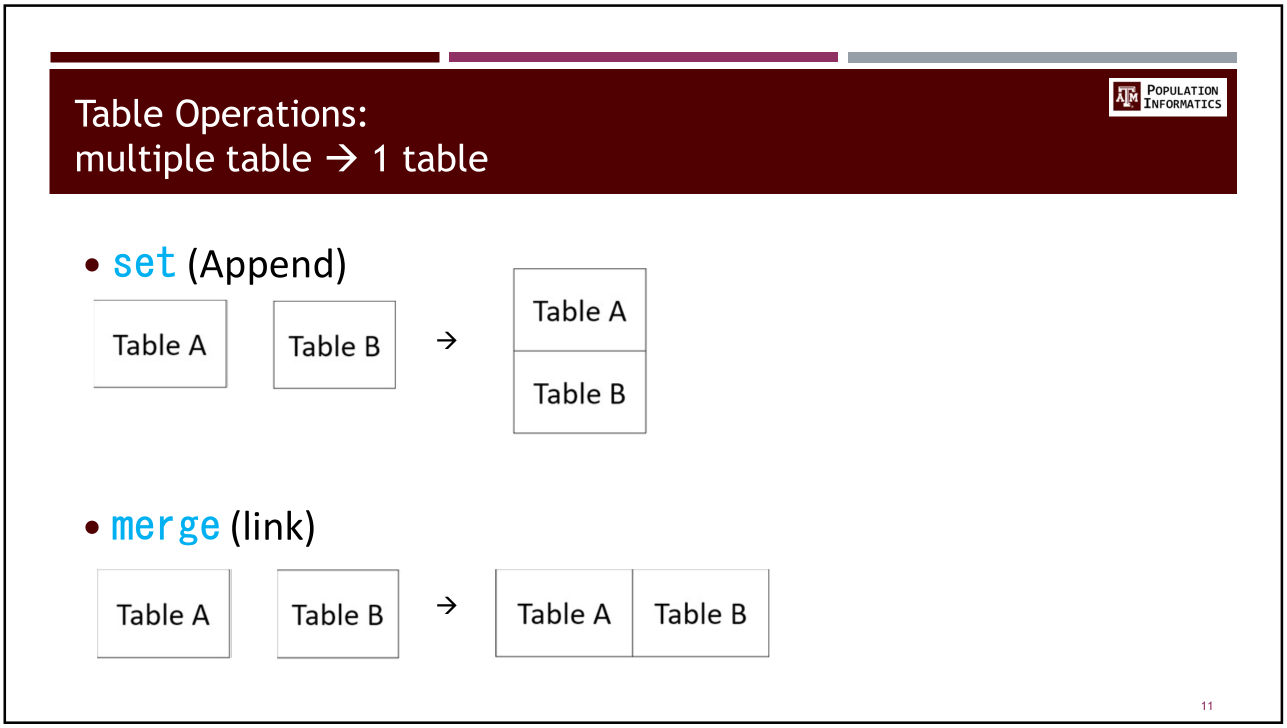


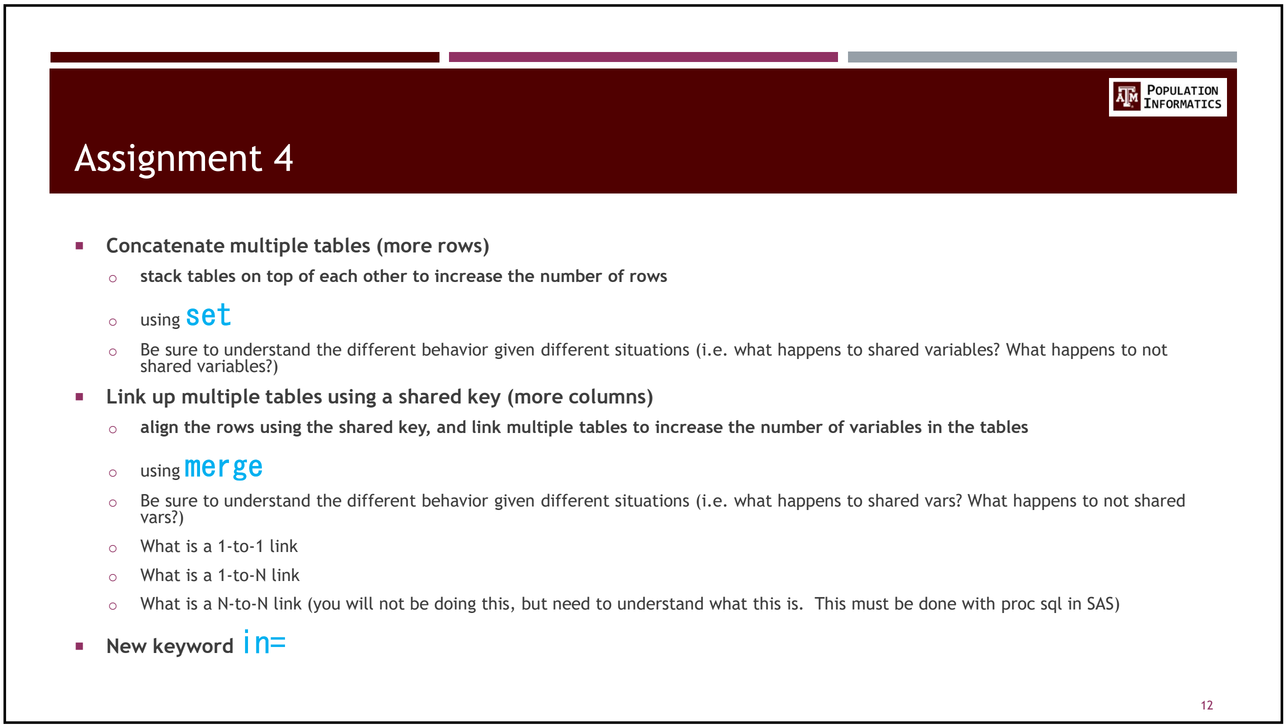
Table Operations: multiple table → 1 table

- set (Append)

Table A Table B → Table A / Table B

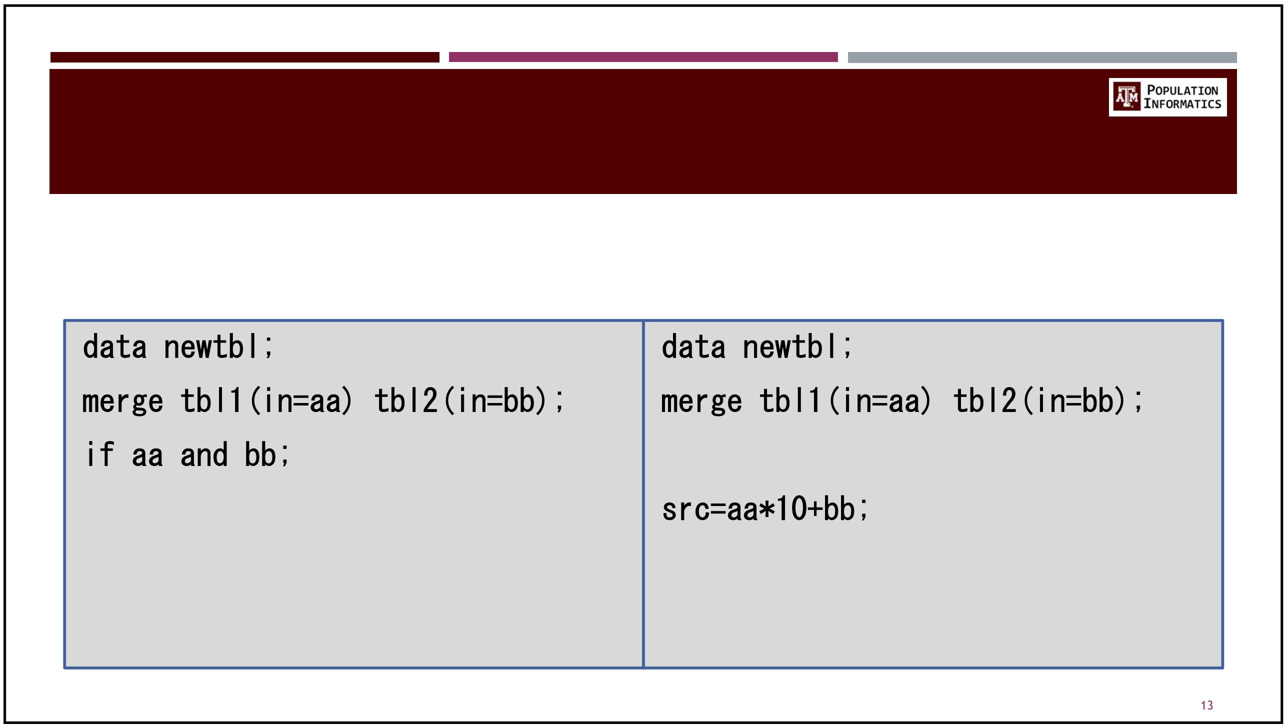
- merge (link)

Table A Table B → Table A Table B



Assignment 4

- Concatenate multiple tables (more rows)
  - stack tables on top of each other to increase the number of rows
  - using `set`
  - Be sure to understand the different behavior given different situations (i.e. what happens to shared variables? What happens to not shared variables?)
- Link up multiple tables using a shared key (more columns)
  - align the rows using the shared key, and link multiple tables to increase the number of variables in the tables
  - using `merge`
  - Be sure to understand the different behavior given different situations (i.e. what happens to shared vars? What happens to not shared vars?)
  - What is a 1-to-1 link
  - What is a 1-to-N link
  - What is a N-to-N link (you will not be doing this, but need to understand what this is. This must be done with proc sql in SAS)
- New keyword `in=`



```
data newtbl;
merge tbl1(in=aa) tbl2(in=bb);
if aa and bb;
```

```
data newtbl;
merge tbl1(in=aa) tbl2(in=bb);

src=aa*10+bb;
```

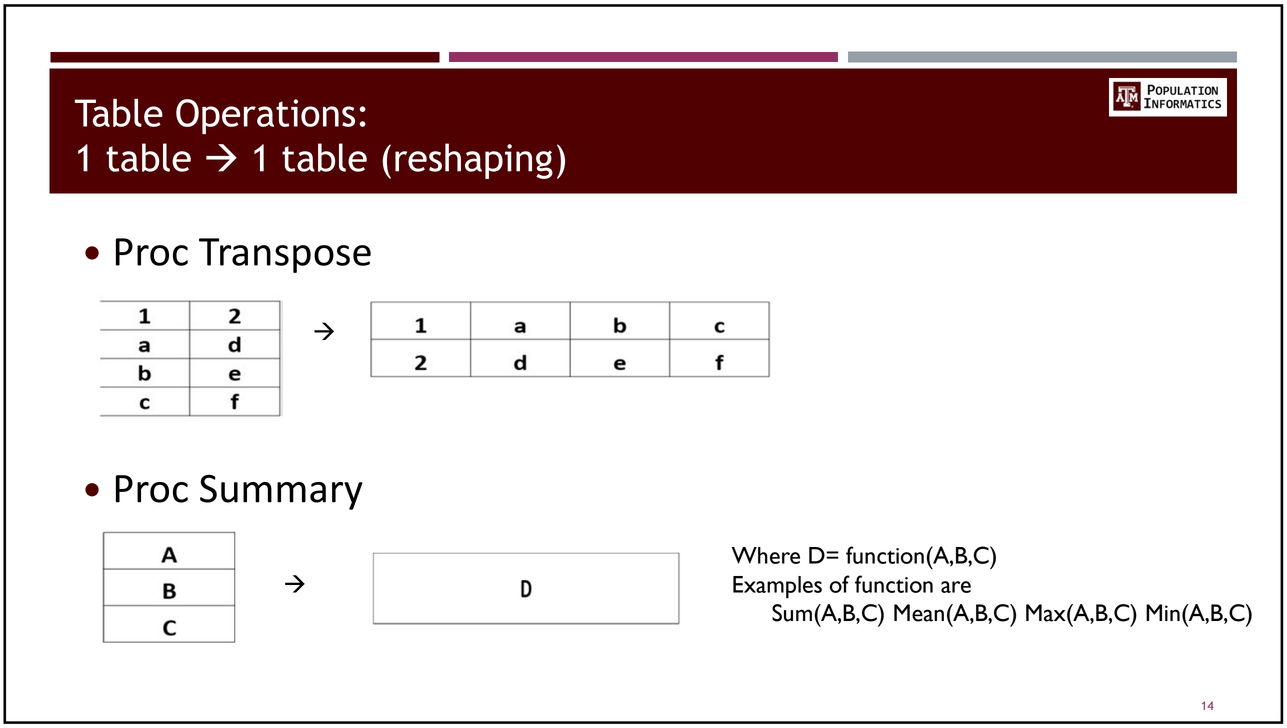


Table Operations:
1 table → 1 table (reshaping)

- Proc Transpose

1	2
a	d
b	e
c	f

→

1	a	b	c
2	d	e	f

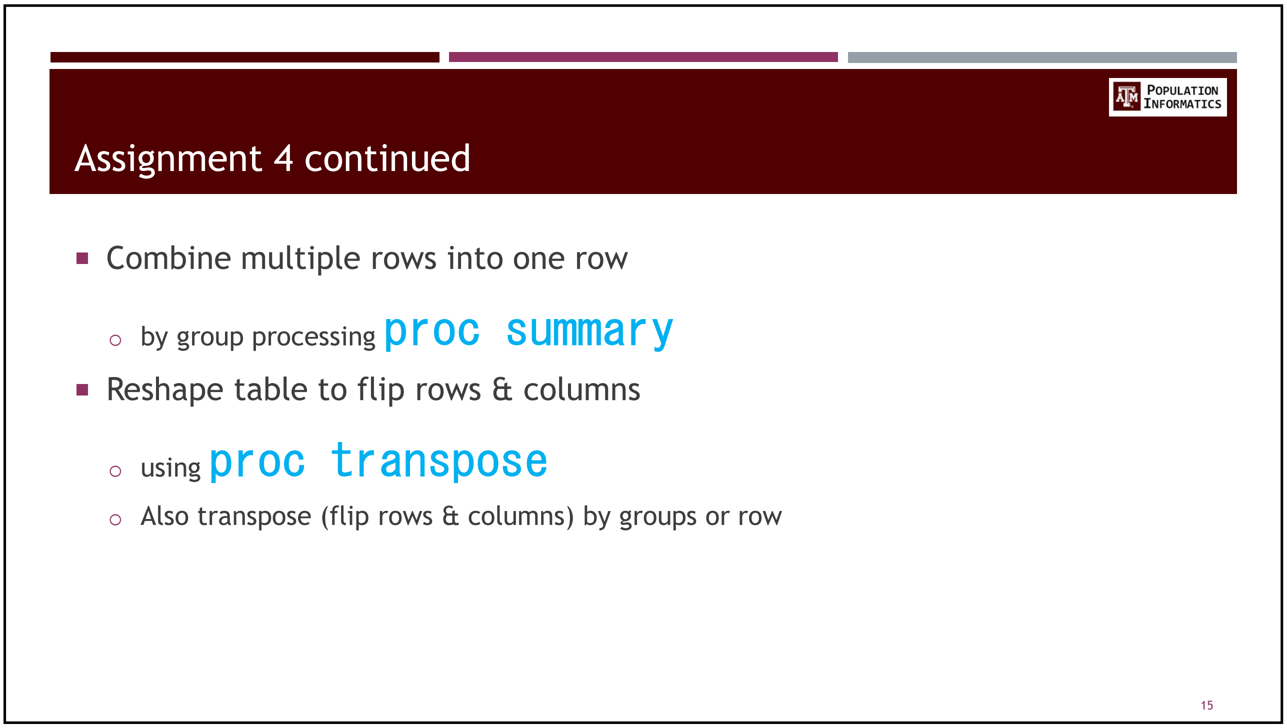
- Proc Summary

A
B
C

→

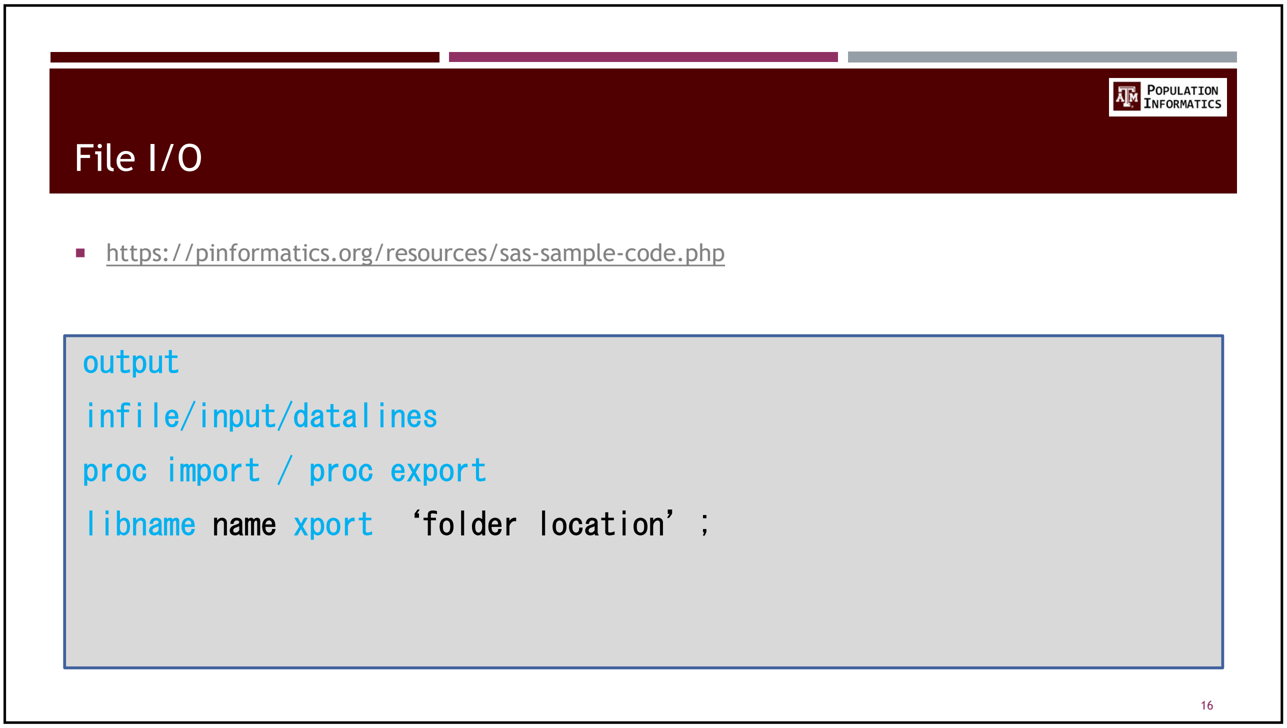
D

Where D= function(A,B,C)
Examples of function are
Sum(A,B,C) Mean(A,B,C) Max(A,B,C) Min(A,B,C)



Assignment 4 continued

- Combine multiple rows into one row
  - by group processing `proc summary`
- Reshape table to flip rows & columns
  - using `proc transpose`
  - Also transpose (flip rows & columns) by groups or row



File I/O

- https://pinformatics.org/resources/sas-sample-code.php

```
output
infile/input/datalines
proc import / proc export
libname name xport  'folder location' ;
```

Population Informatics

demo.sas

17

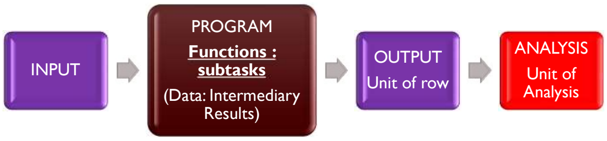
17

UNIT OF ANALYSIS

18

18

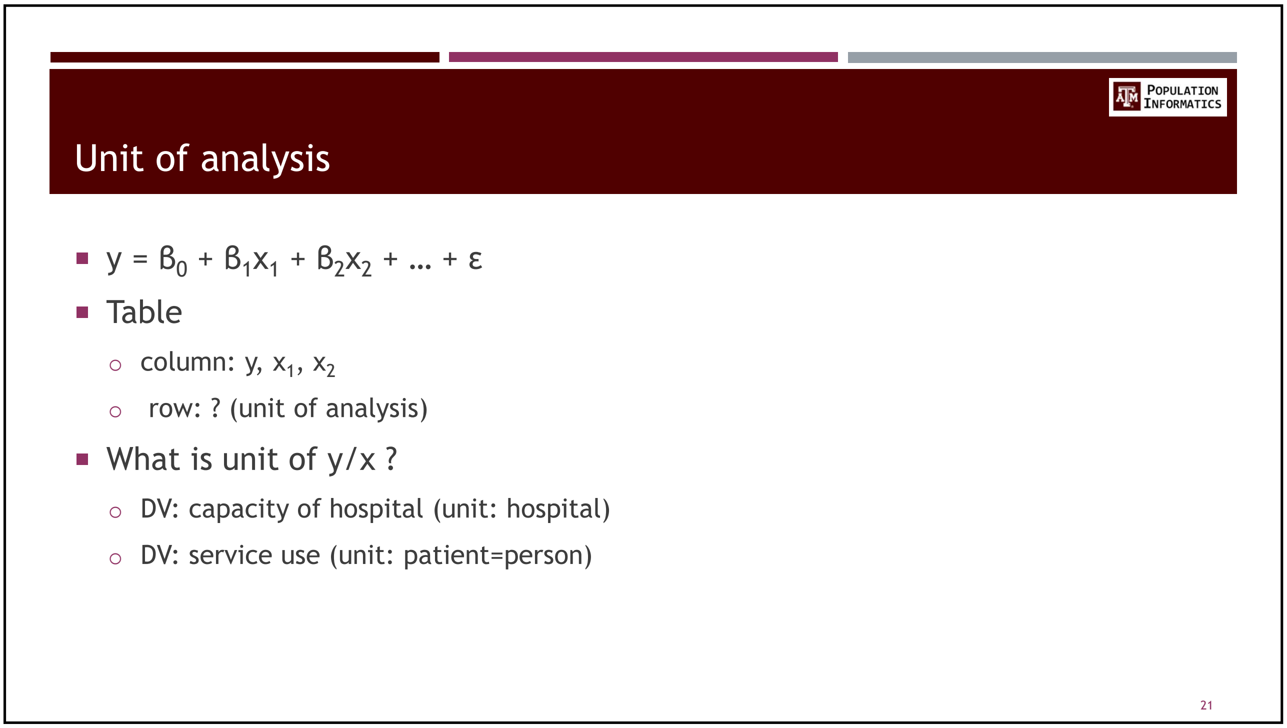
Basic Regression



- $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \varepsilon$
- y: dependent variable
- x_i : independent variables
  - β_i: coefficient
- ε : error term

Unit of analysis

- $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \varepsilon$
- Table
  - column: y, x_1, x_2
  - row: ? (unit of analysis)
- What is unit of y/x ?
  - DV: capacity of hospital (unit: ?)
  - DV: service use (unit: ?)



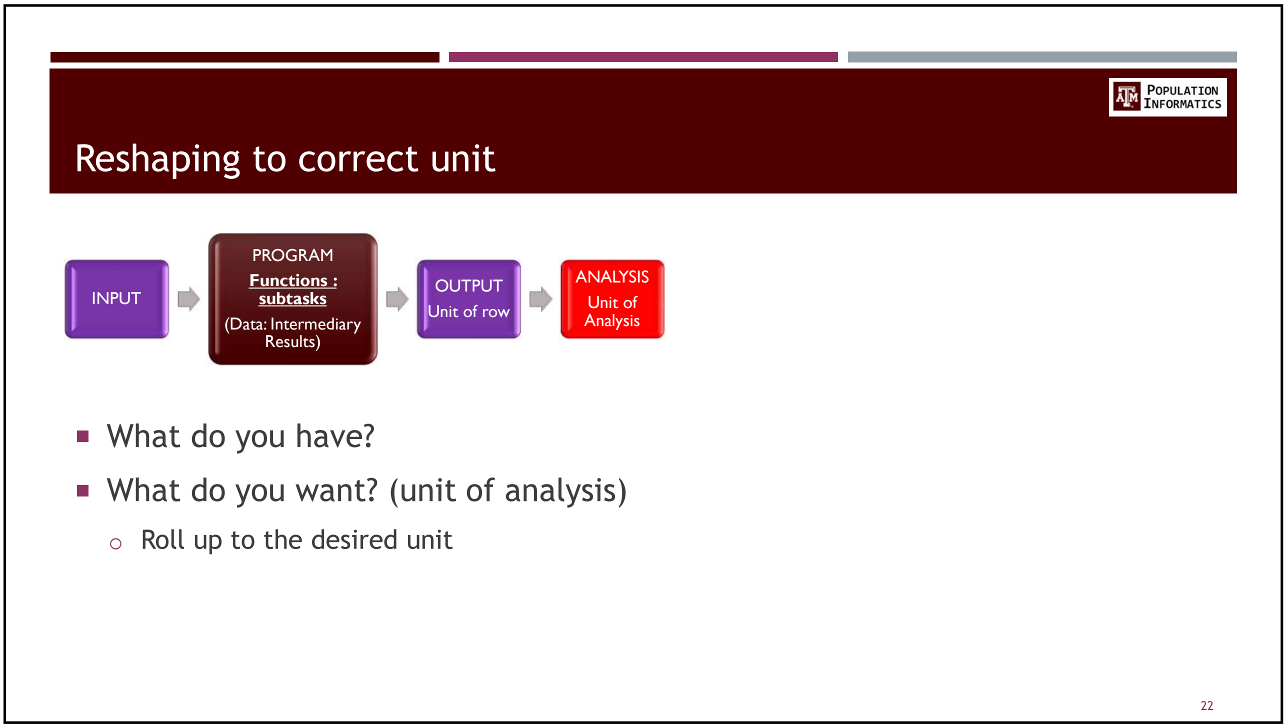
POPULATION INFORMATICS

Unit of analysis

- $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \varepsilon$
- Table
  - column: y, x_1, x_2
  - row: ? (unit of analysis)
- What is unit of y/x ?
  - DV: capacity of hospital (unit: hospital)
  - DV: service use (unit: patient=person)

21

21



POPULATION INFORMATICS

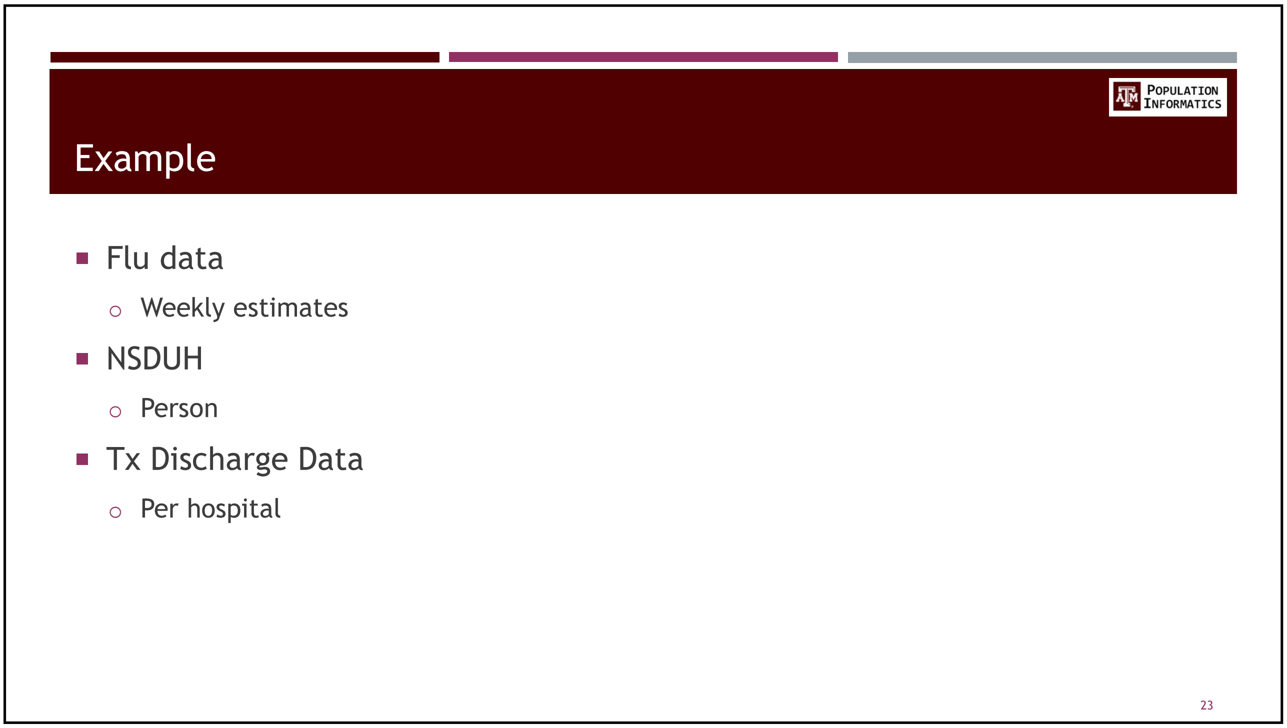
Reshaping to correct unit

INPUT → PROGRAM **Functions : subtasks** (Data: Intermediary Results) → OUTPUT Unit of row → ANALYSIS Unit of Analysis

- What do you have?
- What do you want? (unit of analysis)
  - Roll up to the desired unit

22

22

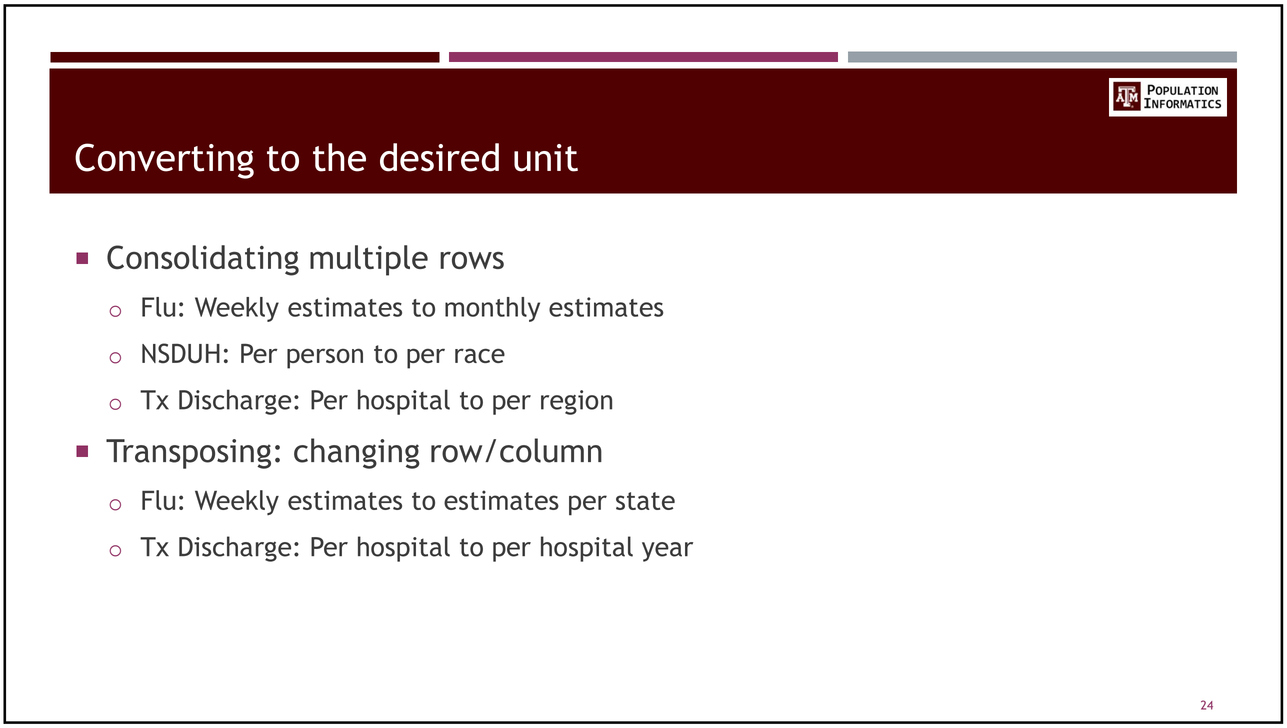


Example

- Flu data
  - Weekly estimates
- NSDUH
  - Person
- Tx Discharge Data
  - Per hospital

23

23

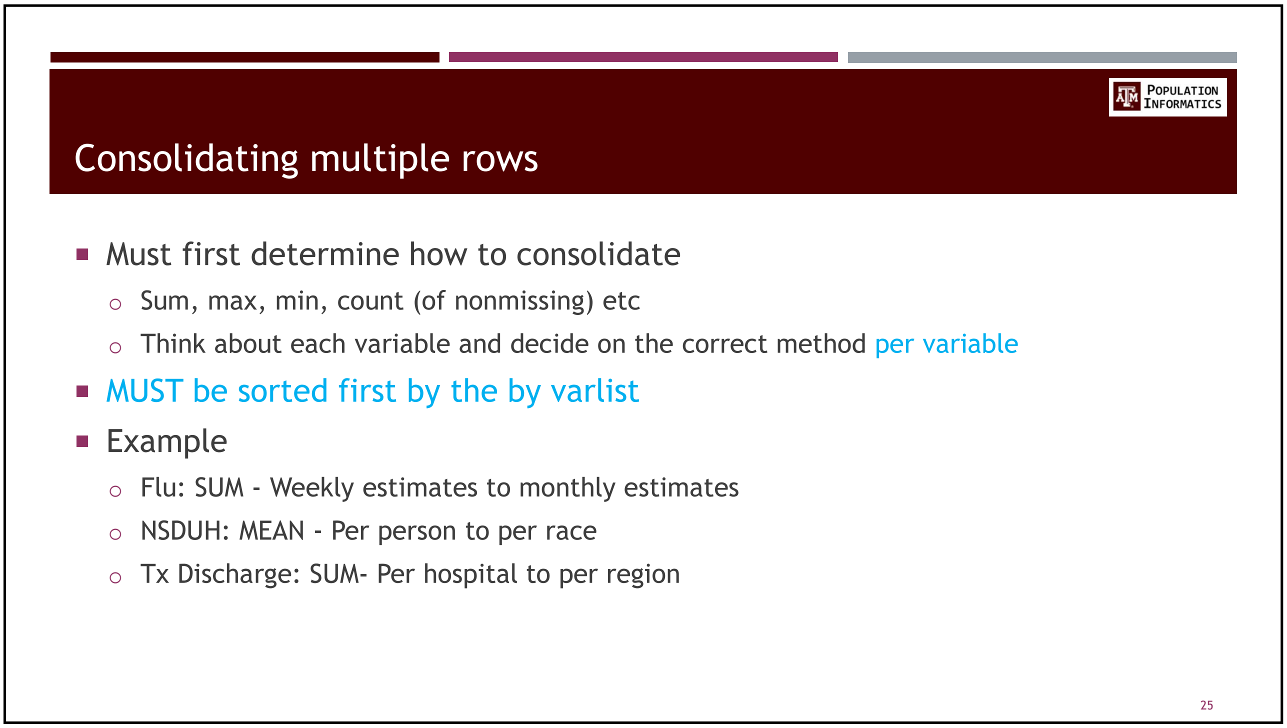


Converting to the desired unit

- Consolidating multiple rows
  - Flu: Weekly estimates to monthly estimates
  - NSDUH: Per person to per race
  - Tx Discharge: Per hospital to per region
- Transposing: changing row/column
  - Flu: Weekly estimates to estimates per state
  - Tx Discharge: Per hospital to per hospital year

24

24

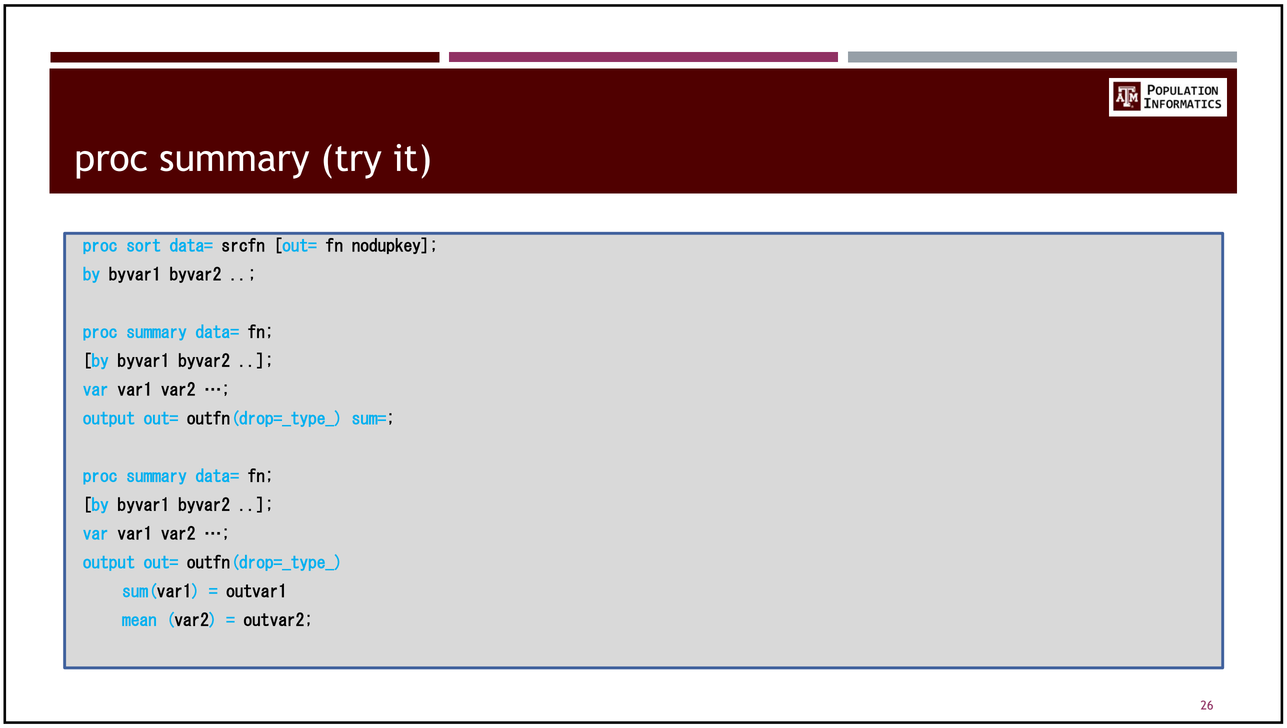


Consolidating multiple rows

- Must first determine how to consolidate
  - Sum, max, min, count (of nonmissing) etc
  - Think about each variable and decide on the correct method per variable
- MUST be sorted first by the by varlist
- Example
  - Flu: SUM - Weekly estimates to monthly estimates
  - NSDUH: MEAN - Per person to per race
  - Tx Discharge: SUM- Per hospital to per region

25

25



proc summary (try it)

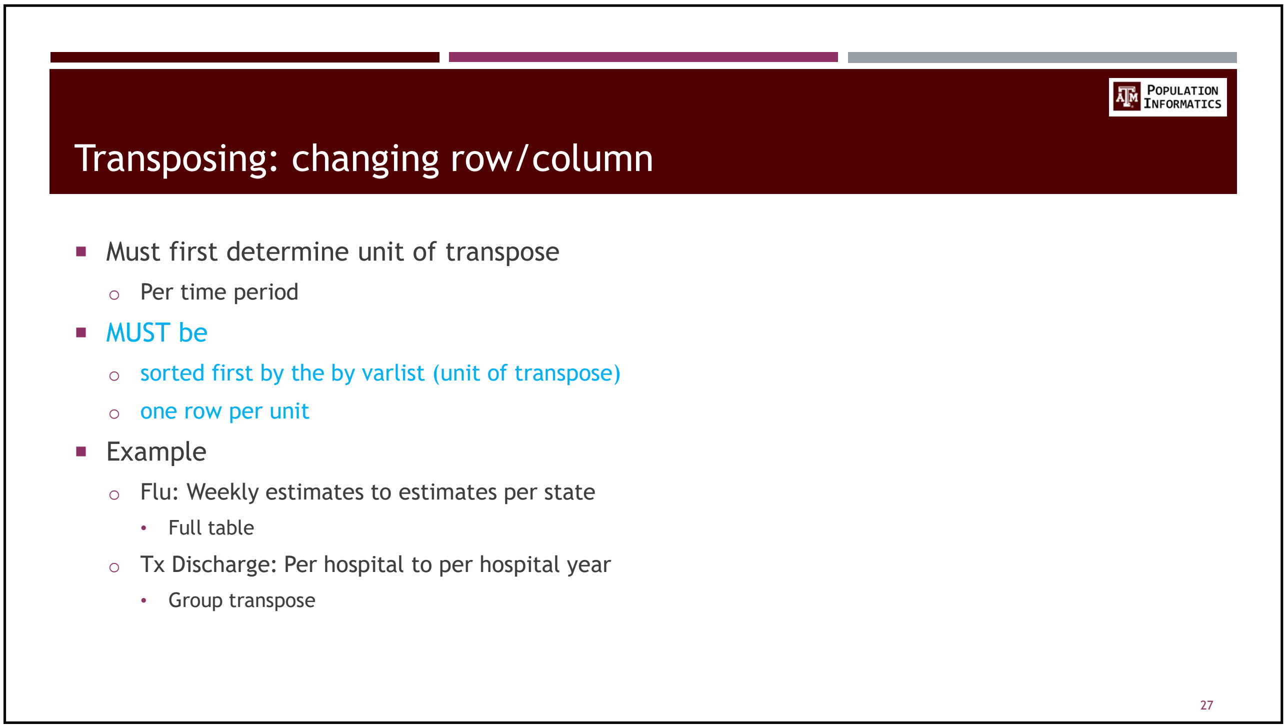
```
proc sort data= srcfn [out= fn nodupkey];
by byvar1 byvar2 ..;

proc summary data= fn;
[by byvar1 byvar2 ..];
var var1 var2 …;
output out= outfn(drop=_type_) sum=;

proc summary data= fn;
[by byvar1 byvar2 ..];
var var1 var2 …;
output out= outfn(drop=_type_)
     sum(var1) = outvar1
     mean (var2) = outvar2;
```

26

26



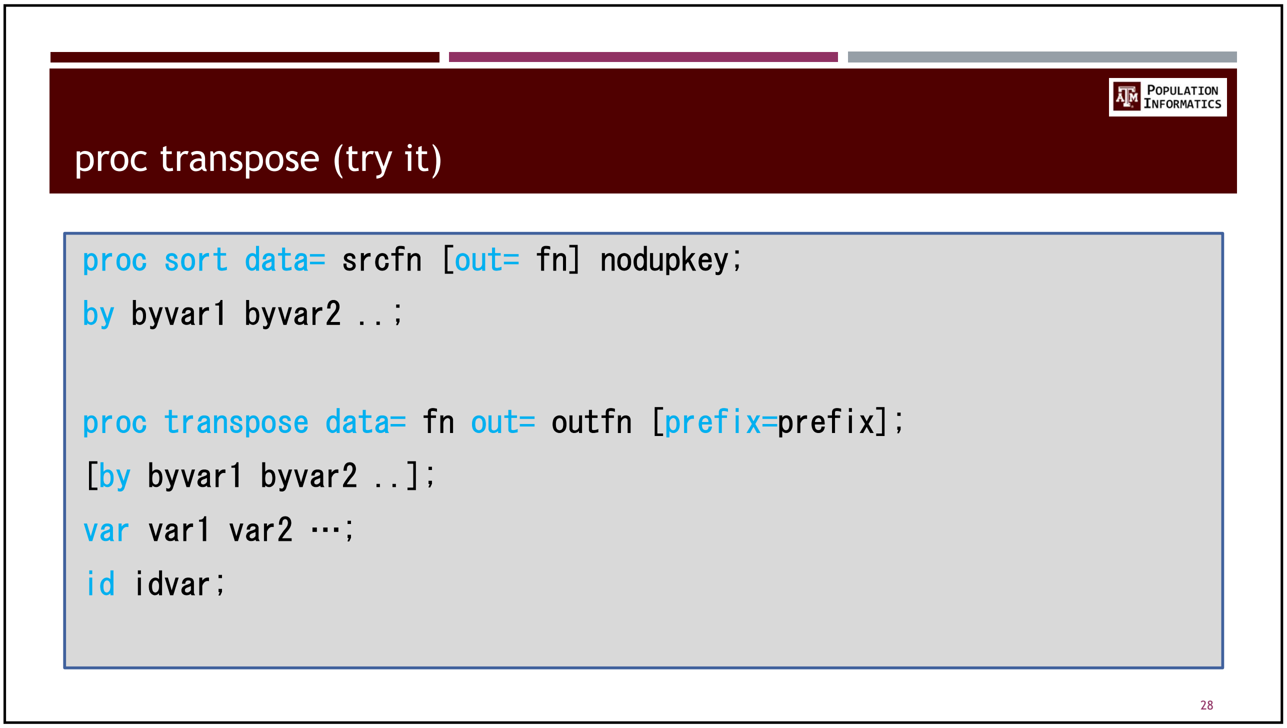
POPULATION INFORMATICS

Transposing: changing row/column

- Must first determine unit of transpose
  - Per time period
- MUST be
  - sorted first by the by varlist (unit of transpose)
  - one row per unit
- Example
  - Flu: Weekly estimates to estimates per state
    - Full table
  - Tx Discharge: Per hospital to per hospital year
    - Group transpose

27

27



POPULATION INFORMATICS

proc transpose (try it)

```
proc sort data= srcfn [out= fn] nodupkey;
by byvar1 byvar2 ..;

proc transpose data= fn out= outfn [prefix=prefix];
[by byvar1 byvar2 ..];
var var1 var2 …;
id idvar;
```

28

28