Reshaping & Combining Tables

Unit of analysis

Combining

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

Reshaping

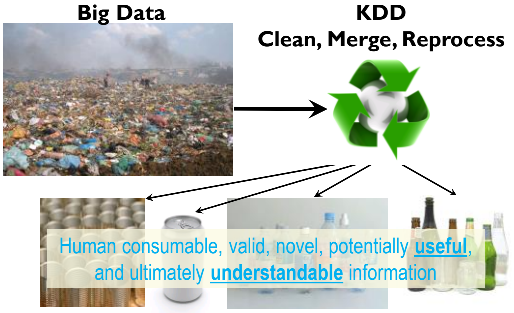
- `proc summary`: consolidate rows
- `proc transpose`: reshape table

Hye-Chung Kum
Population Informatics Research Group
http://research.tamhsc.edu/pinformatics/
http://pinformatics.web.unc.edu/

License:
Data Science in the Health Domain by Hye-Chung Kum is licensed under a
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

Course URL:
http://pinformatics.tamhsc.edu/phpm672

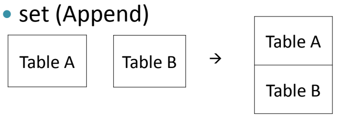
Data Science
Knowledge Discovery & Data mining (KDD)



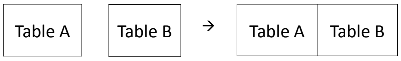
Human consumable, valid, novel, potentially **useful**, and ultimately **understandable** information

Table Operations: multiple table → 1 table

- set (Append)



- merge (link)



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)

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

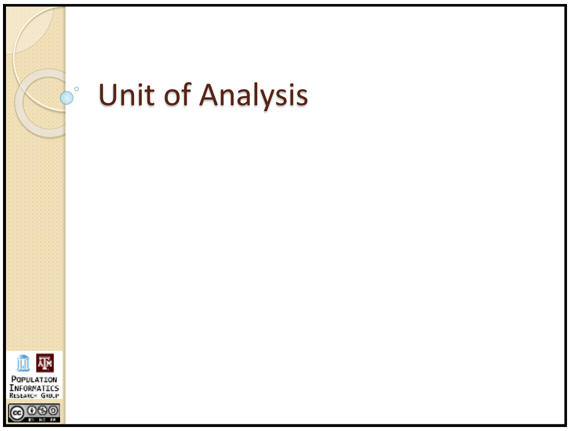
→

D

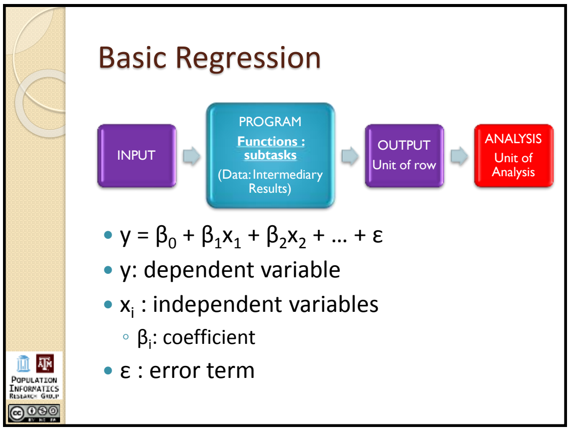
Where D=function(A,B,C)
Examples of function are
Sum(A,B,C) Mean(A,B,C) Max(A,B,C) Min(A,B,C)

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



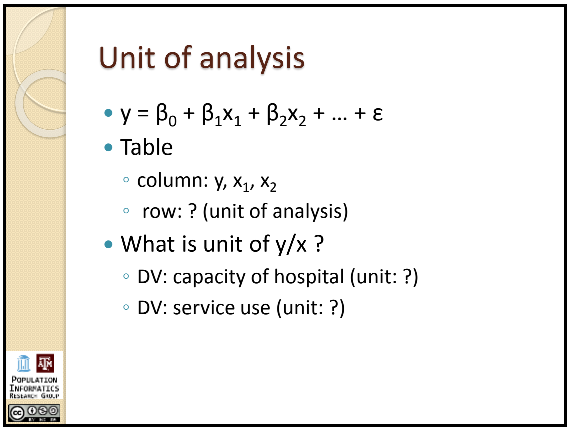
Unit of Analysis



Basic Regression

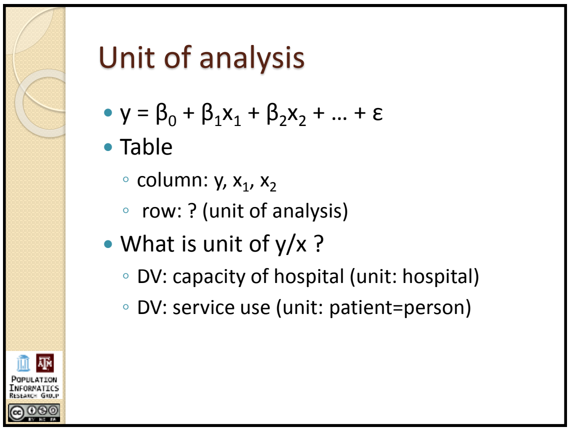
INPUT → PROGRAM **Functions : subtasks** (Data: Intermediary Results) → OUTPUT Unit of row → ANALYSIS Unit of Analysis

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



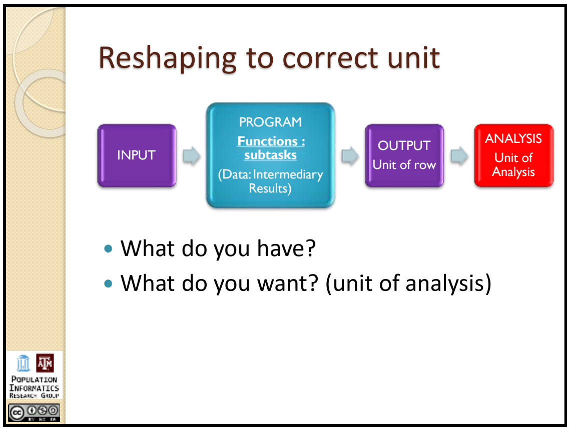
Unit of analysis

- $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \varepsilon$
- 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: ?)



Unit of analysis

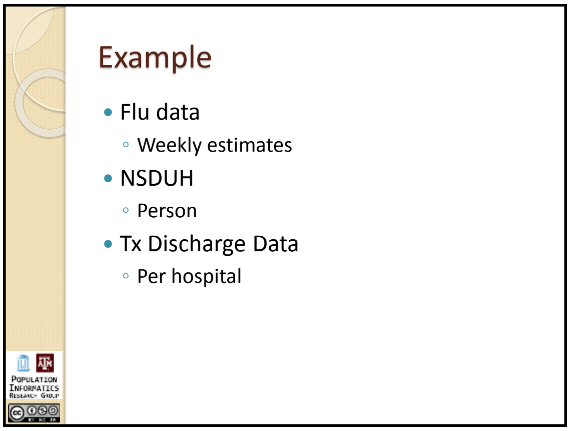
- $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \varepsilon$
- Table
  - column: y, x_1, x_2
  - row: ? (unit of analysis)
- What is unit of y/x ?
  - DV: capacity of hospital (unit: hospital)
  - DV: service use (unit: patient=person)



Reshaping to correct unit

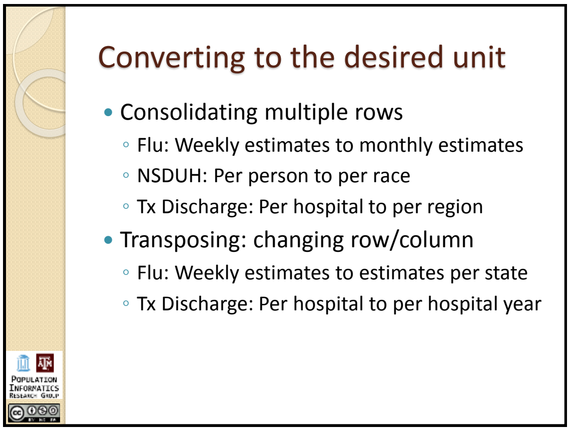
INPUT → PROGRAM **Functions : subtasks** (Data: Intermediary Results) → OUTPUT Unit of row → ANALYSIS Unit of Analysis

- What do you have?
- What do you want? (unit of analysis)



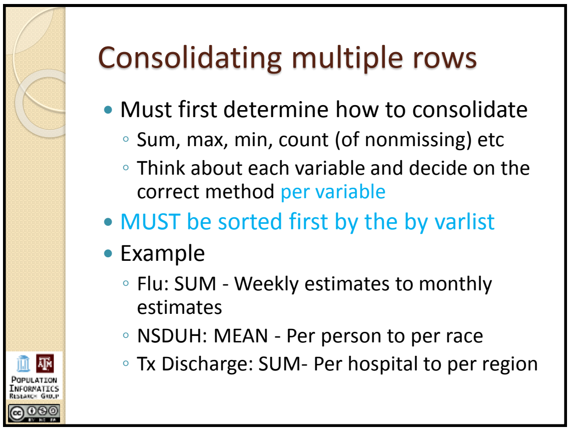
Example

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



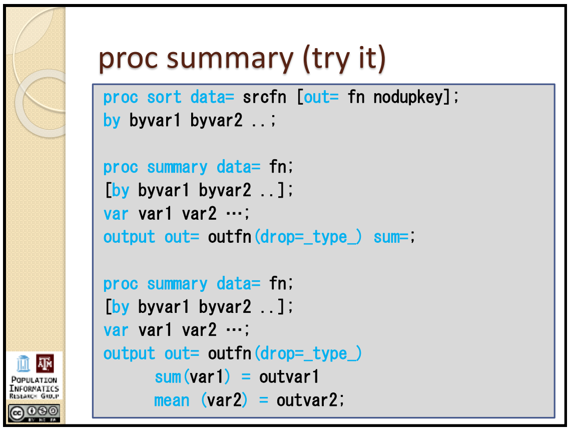
Converting to the desired unit

- 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



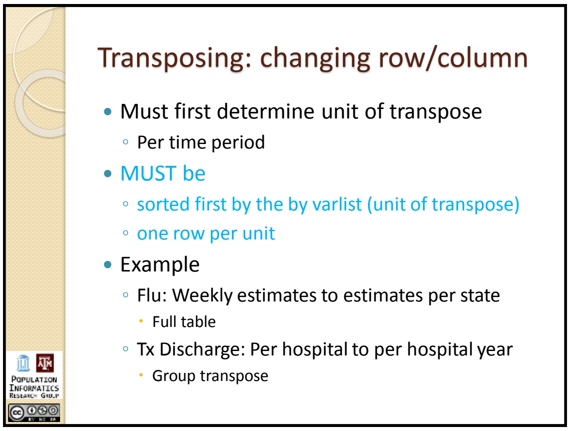
Consolidating multiple rows

- 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



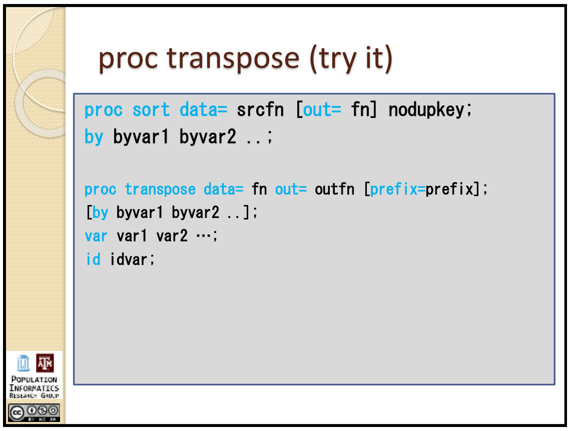
proc summary (try it)

```
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;
```



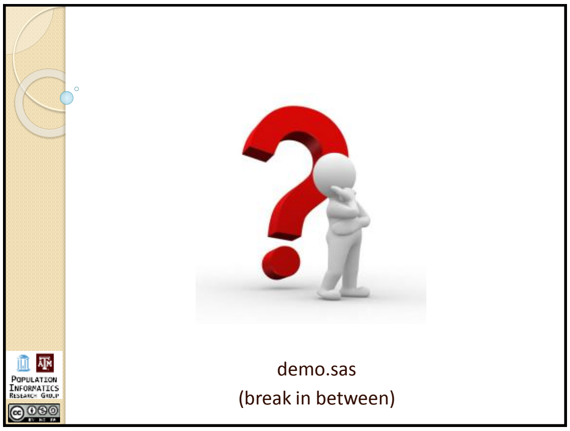
Transposing: changing row/column

- 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



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;
```



demo.sas

(break in between)

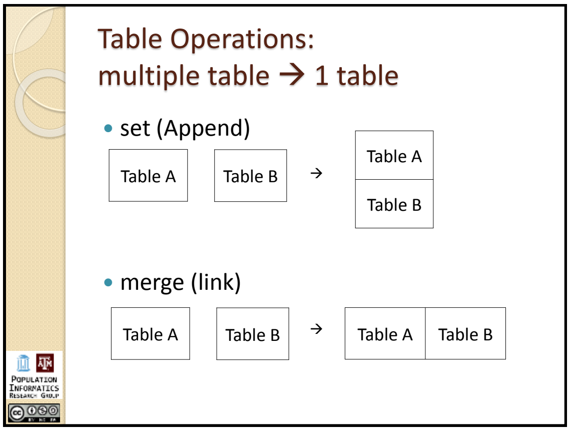


Table Operations: multiple table → 1 table

- set (Append)

Table A Table B →

Table A
Table B

- merge (link)

Table A Table B →

Table A	Table B

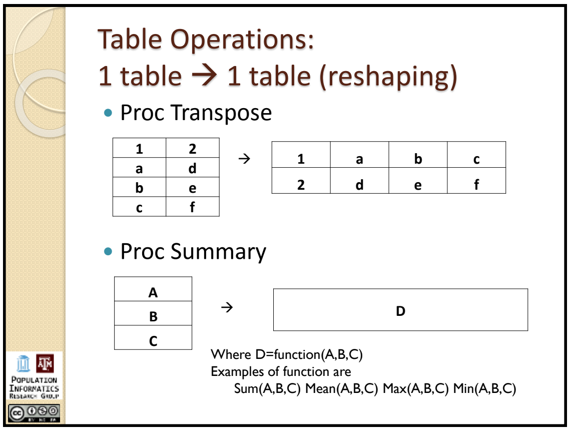


Table Operations: 1 table → 1 table (reshaping)

- Proc Transpose

1	2
a	d
b	e
c	f

→

1	a	b	c
2	d	e	f

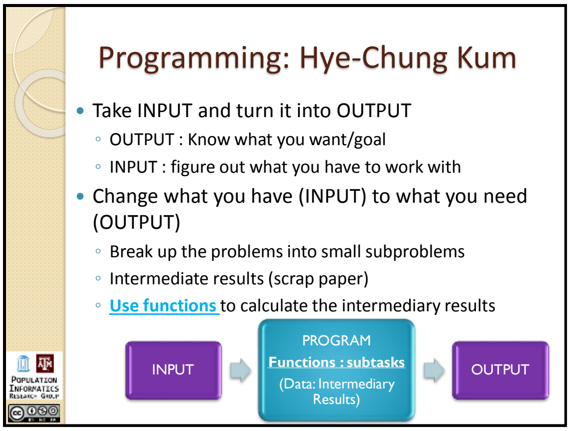
- Proc Summary

A
B
C

→

D

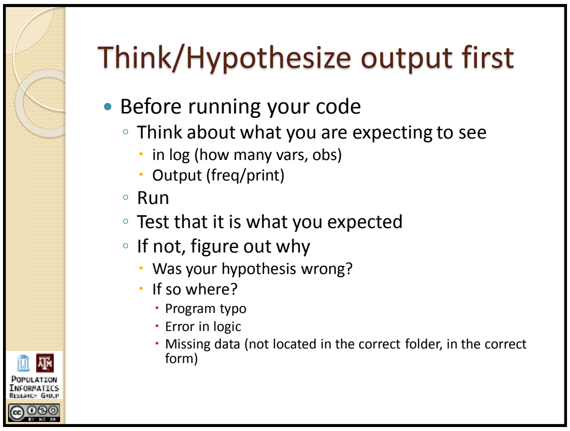
Where D=function(A,B,C)
Examples of function are
Sum(A,B,C) Mean(A,B,C) Max(A,B,C) Min(A,B,C)



Programming: Hye-Chung Kum

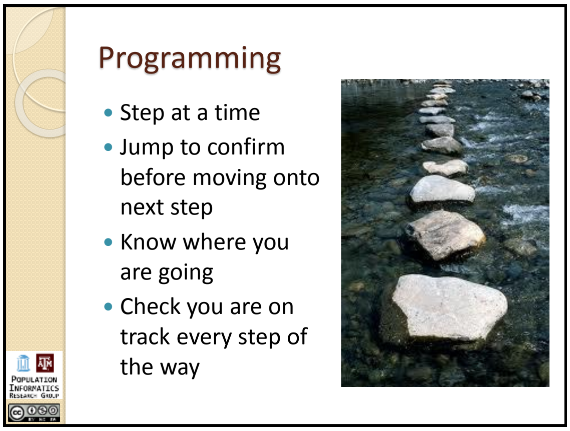
- Take INPUT and turn it into OUTPUT
  - OUTPUT : Know what you want/goal
  - INPUT : figure out what you have to work with
- Change what you have (INPUT) to what you need (OUTPUT)
  - Break up the problems into small subproblems
  - Intermediate results (scrap paper)
  - Use functions to calculate the intermediary results

INPUT
PROGRAM
Functions : subtasks
(Data: Intermediary Results)
OUTPUT



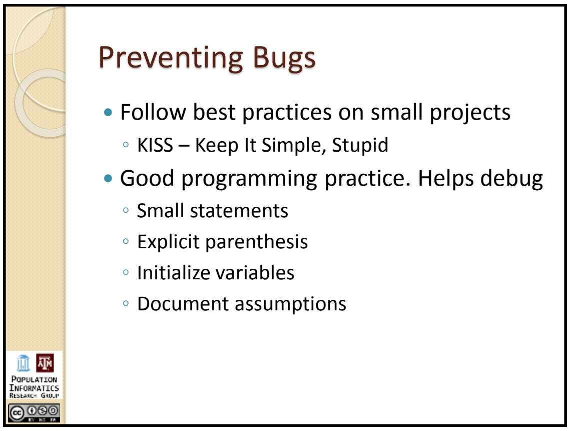
Think/Hypothesize output first

- Before running your code
  - Think about what you are expecting to see
    - in log (how many vars, obs)
    - Output (freq/print)
  - Run
  - Test that it is what you expected
  - If not, figure out why
    - Was your hypothesis wrong?
    - If so where?
      - Program typo
      - Error in logic
      - Missing data (not located in the correct folder, in the correct form)



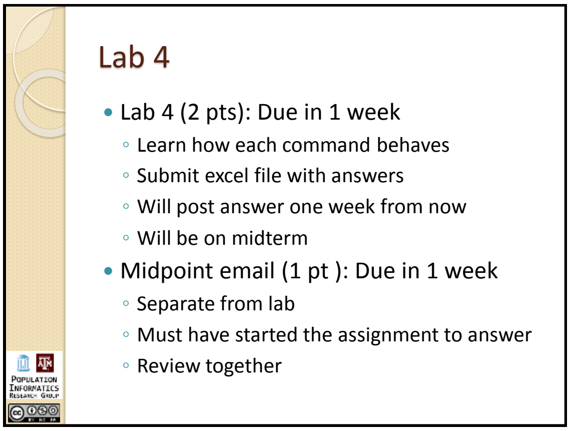
Programming

- Step at a time
- Jump to confirm before moving onto next step
- Know where you are going
- Check you are on track every step of the way



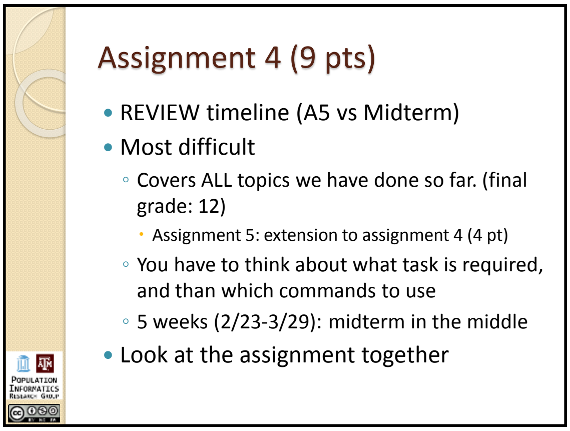
Preventing Bugs

- Follow best practices on small projects
  - KISS – Keep It Simple, Stupid
- Good programming practice. Helps debug
  - Small statements
  - Explicit parenthesis
  - Initialize variables
  - Document assumptions



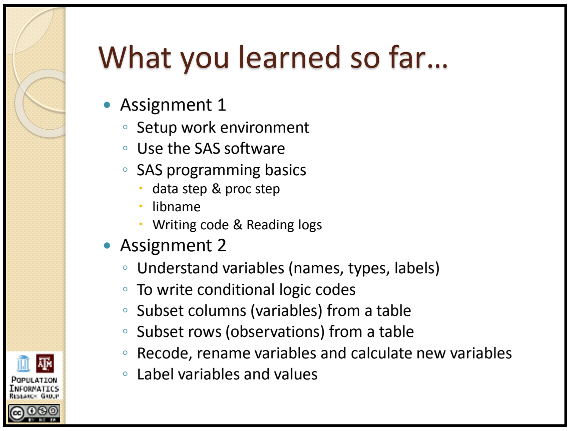
Lab 4

- 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



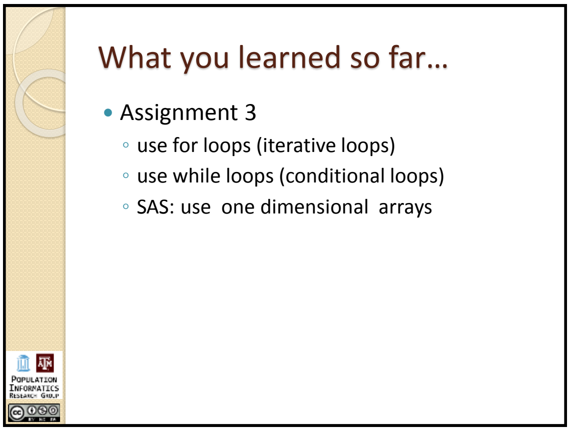
Assignment 4 (9 pts)

- REVIEW timeline (A5 vs Midterm)
- 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 than which commands to use
  - 5 weeks (2/23-3/29): midterm in the middle
- Look at the assignment together



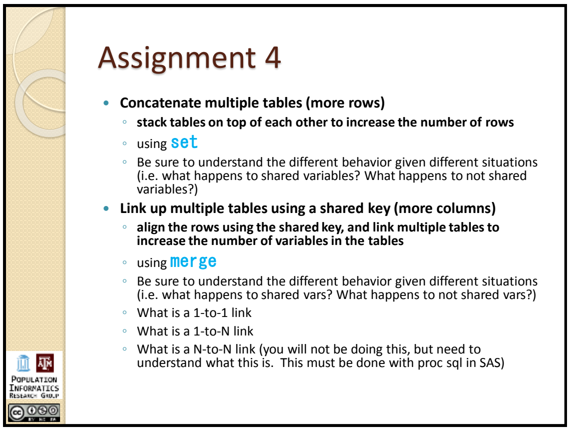
What you learned so far…

- Assignment 1
  - Setup work environment
  - Use the SAS software
  - SAS programming basics
    - data step & proc step
    - libname
    - Writing code & Reading logs
- Assignment 2
  - Understand variables (names, types, labels)
  - To write conditional logic codes
  - Subset columns (variables) from a table
  - Subset rows (observations) from a table
  - Recode, rename variables and calculate new variables
  - Label variables and values



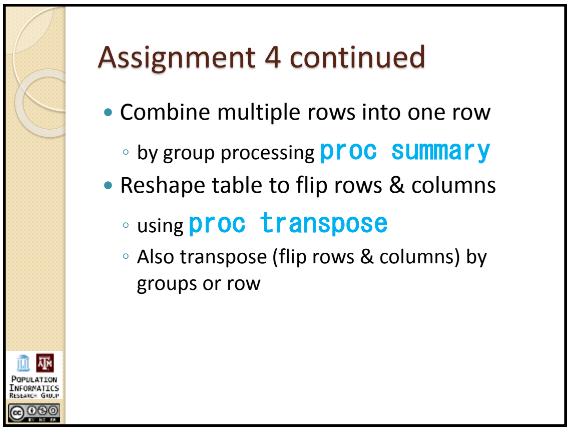
What you learned so far…

- Assignment 3
  - use for loops (iterative loops)
  - use while loops (conditional loops)
  - SAS: use one dimensional arrays



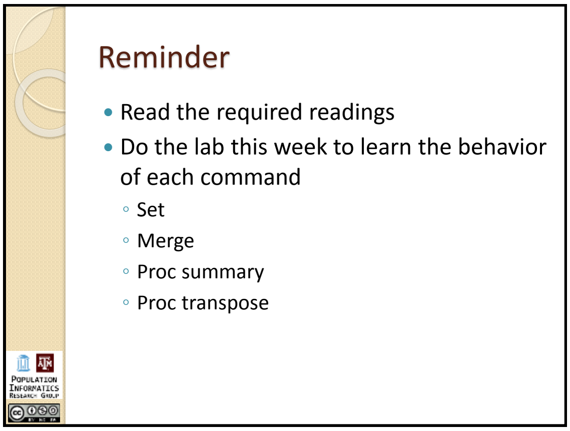
Assignment 4

- **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)



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



Reminder

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