

Creating Loops in STATA

Megha Mehrotra

University of California, San Francisco
Department of Epidemiology and Biostatistics

06 September 2016

Outline

Introduction

Overview

`for` values

`foreach`

Demonstration

Why Loops?

- ▶ Make your code more concise by iterating over many variables
- ▶ Avoid errors!

Why Loops?

- ▶ Make your code more concise by iterating over many variables
- ▶ Avoid errors!

What are some situations where you think loops may be helpful?

Overview

There are 3 basic things you need to tell STATA in order to program a loop:

- ▶ What you want to do repetitively
- ▶ How many times you want to do it
- ▶ What changes each time

forvalues

Basic Syntax:

```
forvalues x = 1/5 {  
  display "`x' + 1 = "  
  display `x' + 1  
  display "  
}
```

- ▶ All the commands between { and } are the commands that are done repetitively

forvalues

Basic Syntax:

```
forvalues x = 1/5 {  
  display "`x' + 1 = "  
  display `x' + 1  
  display "  
}
```

- ▶ `forvalues` tells STATA that you want to do the loop one time for each number between 1 and 5 inclusive.

forvalues

Basic Syntax:

```
forvalues x = 1/5 {  
  display "`x' + 1 = "  
  display `x' + 1  
  display "  
}
```

- forvalues also designates "x" as the placeholder for the thing that will change.

Anywhere that 'x' appears in the command, STATA substitutes the actual value (1, 2, 3, 4, or 5) corresponding with each run of the loop.

foreach

`foreach` allows you to loop through variables in a list.

Example:

Say we have a dataset in which weight is measured in lbs and has been measured monthly. We want to calculate BMI, so we need to convert each weight variable to kgs.

id	<u>wt1</u>	<u>wt2</u>	<u>wt3</u>	<u>wt4</u>	<u>wt5</u>	<u>wt6</u>	<u>wt7</u>	<u>wt8</u>	<u>wt9</u>	<u>wt10</u>	<u>wt11</u>	<u>wt12</u>
1	160	154	153	155	160	163	163	165	166	168	169	168
2	120	122	120	121	125	127	126	126	125	122	123	124
3	143	144	144	145	147	149	150	151	152	152	153	154

foreach

<u>id</u>	<u>wt1</u>	<u>wt2</u>	<u>wt3</u>	<u>wt4</u>	<u>wt5</u>	<u>wt6</u>	<u>wt7</u>	<u>wt8</u>	<u>wt9</u>	<u>wt10</u>	<u>wt11</u>	<u>wt12</u>
1	160	154	153	155	160	163	163	165	166	168	169	168
2	120	122	120	121	125	127	126	126	125	122	123	124
3	143	144	144	145	147	149	150	151	152	152	153	154

Manually: generate kgwt1 = wt1/2.2
generate kgwt2 = wt2/2.2
generate kgwt3 = wt3/2.2
generate kgwt4 = wt4/2.2
generate kgwt5 = wt5/2.2
generate kgwt6 = wt6/2.2
generate kgwt7 = wt7/2.2
... and so forth.

foreach

<u>id</u>	<u>wt1</u>	<u>wt2</u>	<u>wt3</u>	<u>wt4</u>	<u>wt5</u>	<u>wt6</u>	<u>wt7</u>	<u>wt8</u>	<u>wt9</u>	<u>wt10</u>	<u>wt11</u>	<u>wt12</u>
1	160	154	153	155	160	163	163	165	166	168	169	168
2	120	122	120	121	125	127	126	126	125	122	123	124
3	143	144	144	145	147	149	150	151	152	152	153	154

foreach!

```
foreach var of varlist wt1-wt12 {  
  generate kg`var' = `var'/2.2  
}
```

Analyzing Data

Loops are not just for data management or data cleaning, and can be helpful in analysis as well!

- ▶ Describing each variable in a dataset using plots
- ▶ Conducting many pair-wise comparisons
- ▶ Cross tabulating many variables
- ▶ etc...

Coronary Calcium Example

STATA DEMONSTRATION