Towards Self-Contained Data
Attaching Validation Routines to Variables

Bill Rising

Mathematics Department
Bellarmine University

NASUG 2006
Outline

1. Goals
   - Goals for Validation

2. Methods
   - Exploiting Stata
   - Implementation

3. Demo of package
   - Example 1 dataset

4. Finishing Up
   - Questions?
Currently, validation is contained in

- outside documentation
- outside programs (do/ado files)

Can be separated from data too easily
Currently, validation is contained in

- outside documentation
- outside programs (do/ado files)

Can be separated from data too easily
Validation in Dataset

- Currently, validation is contained in
  - outside documentation
  - outside programs (do/ado files)

- Can be separated from data too easily
Currently, validation is contained in
- outside documentation
- outside programs (do/ado files)

Can be separated from data too easily
Validation Persistent

- The validation must follow variables through manipulation
  - Merges
  - Subsetting variables

- Validation rules must be attached to Variables.
Goals for Validation

Validation Persistant

- The validation must follow variables through manipulation
  - Merges
    - Subsetting variables
  - Validation rules must be attached to Variables.
Validation Persistant

- The validation must follow variables through manipulation
  - Merges
  - Subsetting variables

- Validation rules must be attached to Variables.
Validation Persistent

- The validation must follow variables through manipulation
  - Merges
  - Subsetting variables
- Validation rules must be attached to Variables.
Can attach validation by knowing some Stata
- Do not need to know a lot of programming tricks
- Not Easy == Not Used
Validation Easy

- Can attach validation by knowing some Stata
- Do not need to know a lot of programming tricks
- Not Easy == Not Used
Validation Easy

- Can attach validation by knowing some Stata
- Do not need to know a lot of programming tricks
- **Not Easy == Not Used**
Outline

1. Goals
   - Goals for Validation

2. Methods
   - Exploiting Stata
   - Implementation

3. Demo of package
   - Example 1 dataset

4. Finishing Up
   - Questions?
Characteristics

- Characteristics allow attaching most any text to variable or the dataset
- Characteristics follow variables through data manipulations
- Use characteristics!
- Brief review of characteristics (demo)
Characteristics allow attaching most any text to variable or the dataset

Characteristics follow variables through data manipulations

Use characteristics!

Brief review of characteristics (demo)
Characteristics

- Characteristics allow attaching most any text to variable or the dataset
- Characteristics follow variables through data manipulations
- **Use characteristics!**
- Brief review of characteristics (demo)
Characteristics

- Characteristics allow attaching most any text to variable or the dataset
- Characteristics follow variables through data manipulations
- Use characteristics!
- Brief review of characteristics (demo)
Idea: Execute (or Do) Characteristics

- Store validation code in a characteristic
- Write a program to extract the code and execute it
  - The `dochar` command will do this
- Satisfies first and second goals, but not third.
Idea: Execute (or Do) Characteristics

- Store validation code in a characteristic
- **Write a program to extract the code and execute it**
  - The `dochar` command will do this
- Satisfies first and second goals, but not third.
Idea: Execute (or Do) Characteristics

- Store validation code in a characteristic
- Write a program to extract the code and execute it
  - The `dochar` command will do this

Satisfies first and second goals, but not third.
Idea: Execute (or Do) Characteristics

- Store validation code in a characteristic
- Write a program to extract the code and execute it
  - The `dochar` command will do this
- Satisfies first and second goals, but not third.
Outline

1. Goals
   - Goals for Validation

2. Methods
   - Exploiting Stata
   - Implementation

3. Demo of package
   - Example 1 dataset

4. Finishing Up
   - Questions?
Avoiding Hassles - I

- Need to make stored code flexible
  - Renaming variables should not cause problems
  - Code should be rather indifferent to how its results are used
- Would like to extend to use simple things for lists, like Stata's `numlists`
- Need something simple for continuous ranges
- Need to make stored code flexible
  - Renaming variables should not cause problems
  - Code should be rather indifferent to how its results are used
- Would like to extend to use simple things for lists, like Stata’s numlists
- Need something simple for continuous ranges
Avoiding Hassles - I

- Need to make stored code flexible
  - Renaming variables should not cause problems
  - Code should be rather indifferent to how its results are used
- Would like to extend to use simple things for lists, like Stata's numlists
- Need something simple for continuous ranges
Avoiding Hassles - I

- Need to make stored code flexible
  - Renaming variables should not cause problems
  - Code should be rather indifferent to how its results are used

- Would like to extend to use simple things for lists, like Stata’s numlists

- Need something simple for continuous ranges
Avoiding Hassles - I

- Need to make stored code flexible
  - Renaming variables should not cause problems
  - Code should be rather indifferent to how its results are used

- Would like to extend to use simple things for lists, like Stata’s numlists

- Need something simple for continuous ranges
Avoiding Hassles - II

- Would like to avoid using commands altogether, when possible
- Would like to avoid user needing to know details about how the validation works
- Perhaps a dialog box as a front end?
Avoiding Hassles - II

- Would like to avoid using commands altogether, when possible
- Would like to avoid user needing to know details about how the validation works
- Perhaps a dialog box as a front end?
Would like to avoid using commands altogether, when possible
Would like to avoid user needing to know details about how the validation works
Perhaps a dialog box as a front end?
• Dialog box, ckvaredit takes care of attaching the characteristics
• Command ckvar runs through the variables and does the validation
Solution

- Dialog box, `ckvaredit` takes care of attaching the characteristics
- Command `ckvar` runs through the variables and does the validation
How to Enter Validation Rules (Simple)

- For discrete sets of numbers or strings:
  - Set notation works
  - Stata's `numlists` work for numbers

- For continuous ranges of numbers:
  - Set notation works: round brackets ( and ) do not include endpoints, square brackets, [ ] do include endpoints
  - Logic works, using Stata's operators
    - Parentheses do not work, unfortunately
How to Enter Validation Rules (Simple)

- For discrete sets of numbers or strings:
  - Set notation works
  - Stata's `numlist` works for numbers

- For continuous ranges of numbers:
  - Set notation works: round brackets ( and ) do not include endpoints, square brackets, [ ] do include endpoints
  - Logic works, using Stata’s operators
    - Parentheses do not work, unfortunately
How to Enter Validation Rules (Simple)

- For discrete sets of numbers or strings:
  - Set notation works
  - Stata's `numlists` work for numbers

- For continuous ranges of numbers:
  - Set notation works: round brackets `( and )` do not include endpoints, square brackets `[ ]` do include endpoints

- Logic works, using Stata's operators
  - Parentheses do not work, unfortunately
How to Enter Validation Rules (Simple)

- For discrete sets of numbers or strings:
  - Set notation works
  - Stata’s `numlist` works for numbers

- For continuous ranges of numbers:
  - Set notation works: round brackets ( and ) do not include endpoints, square brackets, [ ] do include endpoints
  - Logic works, using Stata’s operators
    - Parentheses do not work, unfortunately
How to Enter Validation Rules (Simple)

- For discrete sets of numbers or strings:
  - Set notation works
  - Stata’s `numlists` work for numbers

- For continuous ranges of numbers:
  - Set notation works: round brackets ( and ) do not include endpoints, square brackets, [ ] do include endpoints
  - Logic works, using Stata’s operators
    - Parentheses do not work, unfortunately
How to Enter Validation Rules (Simple)

- For discrete sets of numbers or strings:
  - Set notation works
  - Stata’s `numlists` work for numbers

- For continuous ranges of numbers:
  - Set notation works: round brackets ( and ) do not include endpoints, square brackets, [ ] do include endpoints

- Logic works, using Stata’s operators
  - Parentheses do not work, unfortunately
How to Enter Validation Rules (Simple)

- For discrete sets of numbers or strings:
  - Set notation works
  - Stata’s `numlist` works for numbers

- For continuous ranges of numbers:
  - Set notation works: round brackets ( and ) do not include endpoints, square brackets, [ ] do include endpoints

- Logic works, using Stata’s operators
  - Parentheses do not work, unfortunately
How to Enter Validation Rules (Complex)

- Use ‘self’ to refer to the variable being checked
- Use ‘valid’ for valid values, and ‘error’ for invalid values
- Avoid branching and looping (though it can be used).
How to Enter Validation Rules (Complex)

- Use ‘self’ to refer to the variable being checked
- Use ‘valid’ for valid values, and ‘error’ for invalid values
- Avoid branching and looping (though it can be used).
How to Enter Validation Rules (Complex)

- Use ‘self’ to refer to the variable being checked
- Use ‘valid’ for valid values, and ‘error’ for invalid values
- Avoid branching and looping (though it can be used).
How to Avoid Reentering Rules

- Can use like `varname` to check just like another variable.
- One big reason for using ‘self’!
How to Avoid Reentering Rules

- Can use like `varname` to check just like another variable.
- One big reason for using ‘self’!
Outline

1. Goals
   - Goals for Validation
2. Methods
   - Exploiting Stata
   - Implementation
3. Demo of package
   - Example 1 dataset
4. Finishing Up
   - Questions?
Looking at the Variables

- *describe* is enough to set up the validation rules
- Ha! How often does that happen?
Looking at the Variables

- `describe` is enough to set up the validation rules
- Ha! How often does that happen?
Entering the Rules

- Type in `ckvaredit`, and work along
- Use the `Reset` button if changes have been saved but do not seem to register
Entering the Rules

- Type in `ckvaredit`, and work along
- Use the **Reset** button if changes have been saved but do not seem to register
Check the Data

- Try `ckvar`
- Drop the error variables, and try `ckvar, total(allerrors)`
- All Done!
Check the Data

- Try `ckvar`
- Drop the error variables, and try `ckvar, total(allerrors)`
- All Done!
Check the Data

- Try `ckvar`
- Drop the error variables, and try `ckvar, total(allerrors)`
- All Done!
Outline

1. Goals
   - Goals for Validation

2. Methods
   - Exploiting Stata
   - Implementation

3. Demo of package
   - Example 1 dataset

4. Finishing Up
   - Questions?
Ask away!