Development Data Boot Camp
String Commands

Ge Sun

University of Notre Dame

May 26, 2023

Introduction

- Numeric variables:
  - You can compare them, add, subtract, multiply and divide them.
  - You can apply command like *sum* to them to get the mean, mode or quantiles.
- String variables:
  - String type refers to variables that are stored as words/sentences.

Introduction

- Why do we need to learn String commands?
  * Some magic commands will help us extract information from string variables.
  * It helps us to make the identifier the same across different datasets.
    - Beth *v.s.* beth
    - South Bend *v.s.* SouthBend *v.s.* Southbend

Command: *split*

Split string variables into two or more parts

- *split var*: var will be split into variable *var1 var2* etc. provided it contains blanks
- *split var, parse(-)*: var will be split using "-" as a separator instead of blanks. Other possible separators: "," "/" "."
- *split var, parse(.) gen(var_part)*: variable *var* will be split into *var_part1 var_part2*, instead of *var1 var2*.
- *split var, destring*: after split, directly replace new string variables with numeric variables for further calculation

Command: *split*

Split string variables into two or more parts

- *split var*: var will be split into variable *var1 var2* etc. provided it contains blanks
- *split var, parse(-)*: var will be split using "-" as a separator instead of blanks. Other possible separators: "," "/" "."
- *split var, parse(.) gen(var_part)*: variable *var* will be split into *var_part1 var_part2*, instead of *var1 var2*.
- *split var, destring*: after split, directly replace new string variables with numeric variables for further calculation

Examples:

- split date 1940/12/10
- split phone number: 1-512-471-3434
- survey questions like: What do you watch on TV? (mark all that apply)

Command: *substr*

- But sometimes, we just need the first few characters or specific subset of characters in the string

Command: *substr*

- But sometimes, we just need the first few characters or specific subset of characters in the string

Syntax of *substr*

substr(s, n_1, n_2)

description: the substring of s, starting at n_1, for a length of n_2

Command: *substr*

- But sometimes, we just need the first few characters or specific subset of characters in the string

Syntax of *substr*

$$\text{substr}(\text{s},\ n_1,\ n_2)$$

description: the substring of s, starting at n_1, for a length of n_2

Examples

$$\text{s} = 1940/12/10,\ n_1 = 1,\ n_2 = 4$$

Command: *substr*

- But sometimes, we just need the first few characters or specific subset of characters in the string

Syntax of *substr*

substr(s, n_1, n_2)

description: the substring of s, starting at n_1, for a length of n_2

Examples

s = 1940/12/10, $n_1 = 1$, $n_2 = 4$

s = 1-512-471-3434, what would *substr(s,2,3)* get?

Command: *substr*

- But sometimes, we just need the first few characters or specific subset of characters in the string

Syntax of *substr*

substr(s, n_1, n_2)

description: the substring of s, starting at n_1, for a length of n_2

Examples

s = 1940/12/10, $n_1 = 1$, $n_2 = 4$

s = 1-512-471-3434, what would *substr(s,2,3)* get?

- Question: Please find the *string_exercise.dta* for our exercise. Use *substr* to extract the day information of variable *Birth_date*

- But there are still cases where we need to extract information from string variables in strange format.

Example:

How could I extract the first three-digit [1] of the following telephone numbers:

773.702.1234
607.255.5241
(212) 992-7042
(212) 854-1754
203-432-4771

[1] Since the first 3 digits is the area code which corresponds to a geographic area like a city

The final tool: Regular expression

- Let's open the Stata official guide about Regular Expressions and learn it!

The final tool: Regular expression

- Let's open the Stata official guide about Regular Expressions and learn it!

Exercise:

- Please use regular expression to identify the telephone number with form as xxx-xxx-xxxx(list)

The final tool: Regular expression

- Let's open the Stata official guide about Regular Expressions and learn it!

Exercise:

- Please use regular expression to identify the telephone number with form as xxx-xxx-xxxx(list)

list tel if regexm(tel," ^[0-9][0-9][0-9]-")

Question: Why are there only three of them?

Get rid of Extra space first!

- *strltrim(s)*
  - Description: *s* without leading blanks (ASCII space character char(32))
  - strltrim(" this") = "this"
- *strrtrim(s)*
  - Description: *s* without trailing blanks
  - Example: strrtrim("this ") = "this"
- *stritrim(s)*
  - Description: *s* with multiple, consecutive internal blanks (ASCII space character char(32)) collapsed to one blank
  - stritrim("hello there") = "hello there"
- *subinstr(s," ","",..)*
  - Description: to remove all blanks in *s*
  - stritrim(" hello there ") = "hellothere"

Get rid of Extra space first!

- *ustrltrim(s)*
  - Description: removes the leading Unicode whitespace characters and blanks from the Unicode string *s*
- *ustrrtrim(s)*
  - Description: remove trailing Unicode whitespace characters and blanks from the Unicode string *s*
  - Domain: Unicode strings
- *ustrtrim(s)*
  - Description: removes leading and trailing Unicode whitespace characters and blanks from the Unicode string *s*

Continue on the exercise:

Exercise:

- Please use regular expression to identify the telephone number with form as xxx-xxx-xxxx(list)

```
replace tel = subinstr(tel," ","",.)
replace tel = ustrtrim(tel)
list tel if regexm(tel," ^[0-9][0-9][0-9]-")
```

Continue on the exercise:

Exercise:

- Please use regular expression to identify the telephone number with form as xxx-xxx-xxxx(list)

```
replace tel = subinstr(tel," ","",.)
replace tel = ustrtrim(tel)
list tel if regexm(tel," ^[0-9][0-9][0-9]-")
```

- Then please identify the telephone number with the form of (xxx)xxx-xxxx and xxx.xxx.xxxx

Do not leave space within your regular expression!

Continue on the exercise:

Exercise:

- Please use regular expression to identify the telephone number with form as xxx-xxx-xxxx(list)

```
replace tel = subinstr(tel," ","",.)
replace tel = ustrtrim(tel)
list tel if regexm(tel,"^[0-9][0-9][0-9]-")
```

- Then please identify the telephone number with the form of (xxx)xxx-xxxx and xxx.xxx.xxxx

Do not leave space within your regular expression!

Now we can identify them, then how could we change them into some unified format?

Command: *regexs*

Let's go back to the Stata official guide and learn this command from the example!

Command: *regexs*

Let's go back to the Stata official guide and learn this command from the example!

Exercise

- Change the telephone number all into xxx-xxx-xxxx

Command: *egen,concat*

- Another way to put different parts of strings together.
- Try:

```
split tel_new, parse(-)
egen tel_regenerated = concat(tel_new1 tel_new2 tel_new3)
gen str link = "-"
egen tel_regenerated2 = concat(tel_new1 link tel_new2 link
tel_new3)
```

Capitalize or not?

- *strupper(s)*
  - Description: uppercase ASCII characters in string *s*.
  - Example: strupper("this") = "THIS"
- *strlower(s)*
  - Description: lowercase ASCII characters in string *s*.
  - Example: strlower("THIS") = "this"
- *strproper(s)*
  - Description: a string with the first ASCII letter and any other letters immediately following characters that are not letters capitalized; all other ASCII letters converted to lowercase. Unicode characters beyond ASCII are treated as characters that are not letters.
  - Example:
    - strproper("mR. joHn a. sMitH") = "Mr. John A. Smith"
    - strproper("jack o'reilly") = "Jack O'Reilly"
    - strproper("2-cent's worth") = "2-Cent'S Worth"