

Stata Commands

5/23/2018

Clearing All Data

```
clear all
```

Clearing Output Screen

```
cls
```

Log files

```
log using filename.ext, replace  
log using filename.ext, append  
log close filename.ext  
log close all
```

Variables

```
vblname[_n-1] // Lagged variable  
  
gen obsnbr = _n // Generate observation number  
  
sort vbl1 vbl2 ... // Sort variables: first by vbl1 then vbl2  
  
mkspline vblnew0 scalarbreak1 vblnew1 scalarbreak2 vblnew2 = vblold // Linear spline  
  
mvencode vblname, mv(-9999) // Change vblname missing values to -9999  
mvdecode vblname, mv(-9999) // Change vblname -9999's to missing values  
mvencode _all, mv(-9999) // Change all missing values to -9999  
mvdecode _all, mv(-9999) // Change all -9999's to missing values  
  
capture confirm variable vblname // Does variable exist?  
if _rc {  
    // variable does not exist  
}  
  
list vblname in 1/20 // List observations from 1 to 10
```

Temporary Names

```
tempname tmp // Define tmp  
Reference to: `tmp' // Precede the temporary name with a grave accent and follow it with a single quote
```

Scalars

```
scalar scalarname = 10 // Define scalar  
  
capture confirm scalar scalarname // Does scalar exist?  
if _rc {  
    // scalar does not exist  
}
```

Manipulating strings

```
substr(str, istart, ilength)
```

Listing all Categories for a Category String Variable

```

encode vblnamestr, generate(vblcategory)
label list vblcategory
label drop vblcategory
drop vblcategory

```

Defining Labels for Categorical Data Variable

```

label define vblcategory -1 "NA" 0 "CatName0" ...
// Specify mapping from category name strings
// to the integers representing the category names

```

Encoding Categorical String Data Variable

```

encode vblnamestr, generate(vblcategory)

```

Using Dummy Variables

```

regres vblnamedep i.vblcategory
regres vblnamedep i. vblcategory , allbaselevels // Reports the base category
regres vblnamedep ib3.vblNameExp i.vblcategory, allbaselevels // Reports the base category (3)

```

Using Interactive Variables from Integer Variables

```

regres vblnamedep i.vblcategory1#i.vblcategory2, allbaselevels

```

Adding Labels to Integer Variables

```

label values vblnameint vblnamecatlabels

```

Cutpoints and Table Commands

```

egen vblnamecat = cut(vblname), at(-10, 0, 10, 20, ...) // vblnamecat = -10 or 0 or 10 or 20, ...
egen vblnamecat = cut(vblname), at(-10, 0, 10, 20, ...) icodes // vblnamecat = 0, 1, 2, 3, ...

table vblnamecat, contents(n vblname min vblname max vblname ...)
tabulate vblnamecat

```

Creating Dummy Variables at Cutpoints for Continuous Data

```

egen temp = cut(vblnamecont), at(0, 1, 3) icodes
gen dum_0to1 = temp == 0
gen dum_1to3 = temp == 1

```

Suppressing Stata commands with display

```

quietly {
    noisily display "Test line 1"
    noisily display "Test line 2"
    noisily display "Test line 3"
    noisily display "Test line 4"
}

```

DOS Commands

```

!rename oldname.ext newname.ext

```

Mac Command: Change Personal ado Directory

```

sysdir set PERSONAL /Users/fwesthoff/Documents/ado/personal/

```

Constraints

```
constraint 1 vblname1 = vblname2
constraint 2 vblname3 = vblname4
regress ..., constraint(1)
constraint drop _all
```

Weighted regression

```
[aweight = 1/vblname]
```

Wald tests

```
test (vblname1 == vblname2)
```

Predictions

```
regress depvbl expvbl if sample
predict preddepvbl // Predicts values for all obs
predict preddepvbl if (sample) // Predicts values for in sample obs

probit depvbl expvbl
predict phat // Predicts probabilities
predict predz, xb // Predicts z's
```

Regression Output

```
ssc install outre2 // This is done once to download the code
```

```
outre2 using "C:\Users\fwesthoff\Documents\misc\Research\PitchFX\Results.doc", replace stats(coef tstat)
outre2 using "C:\Users\fwesthoff\Documents\misc\Research\PitchFX\Results.doc", append stats(coef tstat)
```

```
outre2 using "C:\Users\fwesthoff\Documents\misc\Research\PitchFX\Results.txt", replace stats(coef tstat)
outre2 using "C:\Users\fwesthoff\Documents\misc\Research\PitchFX\Results.txt", append stats(coef tstat)
```

Displaying vectors and matrices – A vector is stored as a matrix

```
tempname tempmatrixname // Define temporary matrix name
matrix `tempmatrixname' = e(b) // NB: The single quote (') is NOT a "smart" single quote (').
matrix list `tempmatrixname' // NB: The single quote (') is NOT a "smart" single quote (').

matrix matrixname = e(b) // Define permanent matrix name
scalar element = matrixname[1,1] // Extract element in row 1 column 1
```

Merge dta files

```
use mainfile, clear
merge 1:1 vblmatch using supplementaryfile.dta
```

```
use mainfile, clear
merge m:1 vblmatch using supplementaryfile.dta
```

Create txt file

```
file open statafilename using "file.txt", write replace
file write statafilename "Ascii string 1" _tab "Ascii string 2" _tab (scalar) _n // Simple output
file write statafilename "Ascii string 1" _tab "Ascii string 2" _tab %7.5f(scalar) _n // Fancy output
file close statafilename
file close all
```

Deleting a file

```
erase filename.ext
```

Loops

```

foreach name in a b {
    display "`name'"
}

local namelist "a b"
foreach name of local namelist {
    display "`name'"
}

levelsof vblname, local(vblnamelist)
foreach name of local vblnamelist {
    display "`name'"
}

forvalues index = istart (increment) istop { // istart, increment, and istop must be local vbls or numbers
    display `index' // index is a tempname
}
// NB: (increment) is NOT optional. For example, forvalues index = 1 (1) 5 {

```

Loops and Regression

```

levelsof vblname, local(vblnamelist)
foreach name of local vblnamelist {
    display "`name'"
    regress vblDep vblExp if vblname == "`name'"
}

forvalues i = 1 (1) 5 {
    regress vblDep vblExp if vblcategory == `i'
}

```

Executing do Files

```

do filename.do // Executes the file and displays output
run filename.do // Executes the file without displaying output
do filename.do, nostop // Continues execution after encountering an error
run filename.do, nostop // Continues execution after encountering an error

```

Do file arguments

```
do dofile.do arg1 arg2 ...
```

File dofile.do

```

display `1'
display `2'
display `3'

```