

How do I read data into Stata?

Vincent Kang Fu

Department of Sociology



15 February 2006

Introduction

Use online resources

<http://www.ats.ucla.edu/stat/stata/default.htm>

<http://www.stata.com/support/faqs/>

Read the manual: `-infile-`, `-infix-`, `-insheet-`

To do serious work with any statistical package
the documentation is indispensable

Stata manuals have plenty of examples

Find examples or models to follow

<http://www.ipums.org> provides Stata code for
reading in data

After you've done this once, keep your Stata code
handy and modify it to fit new situations

It's really not that difficult, but you have to take the initiative—there are plenty of resources out there

Unless you've made a deal with the devil, it will not work right the first time—don't get discouraged

Fixed format data

Sometimes you have fixed format data:

```
19893601010770102211111111104642434242431101111111110965233423233101
1989360101287010110122222330843457632456110122222330960284528283117
198936010114701011011111111108682034202031101111111110773151217151105
19893601012320101101222223312543455303441101221111110757315530315103
19893601011470101101112222331043457632456110111332330234547632546209
1989360101287010110111222330248406531405110111222330255335530334107
198936010119501022111111111104691923191922211111111111269192319192100
19893601012760101101112223307553355303341101111111111259294529294104
1989360101033010110111332331119699734578110111342330126638633567106
1989360101217010221111222330752366531365110111222331160284528283108
1989360101147010110111222331138507632506110111222330148416531415109
19893601012170101101112223306533565313551101111111110457315530315104
1989360101136010110111332330962264526263110111222331264243424242102
1989360101276010110111222331259294529293110111332330364243424242105
```

Marriage license data from the National Center for Health Statistics

With fixed format data you'd better have some documentation accompanying your data

MARRIAGE DETAIL RECORD

Tape Location	Field Name	Item and Code Outline
1-4	4	[DATAYEAR] Year of Marriage
		1989 ... 1989
		1990 ... 1990
		1991 ... 1991
		1992 ... 1992
		1993 ... 1993
		1994 ... 1994
		1995 ... 1995

5 1 [REGNOCC]

Region of Marriage

6 1 [DIVNOCC]

Division of Marriage

Region is in position 5 and Division is in 6.

00 ... POSSESSIONS

1 ... NORTHEAST

 1 ... New England

 2 ... Middle Atlantic

2 ... MIDWEST

 3 ... East North Central

 4 ... West North Central

3 ... SOUTH

 5 ... South Atlantic

 6 ... East South Central

 7 ... West South Central

4 ... WEST

 8 ... Mountain

 9 ... Pacific

7-8 2 [STATEOCC] State of Marriage

This field contains codes for States that have been admitted to the Marriage Registration Area (MRA). State names, codes, and year of admission to the MRA are:

(1)

Tape Location	Field Name	Item and Code Outline	(Continued)
		01	... Alabama 1957
		02	... Alaska 1957
...			
		50	... Wisconsin 1957
		51	... Wyoming 1957
		52	... Puerto Rico 1957
		53	... Virgin Islands 1957

Tape Location	Field Name	Item and Code Outline	(Continued)
9-13	5	DATE OF MARRIAGE	
9-10	2	[MARMON]	Month of Marriage
		01	... January
		02	... February
		03	... March
		04	... April
		05	... May
		06	... June
		07	... July
		08	... August
		09	... September
		10	... October
		11	... November
		12	... December
11-12	2	[MARDAY]	Day of Marriage
		01-31	... As applicable to month of marriage

13 1 [WEEKDAY] Day of Week of Marriage

- 1 ... Sunday
- 2 ... Monday
- 3 ... Tuesday
- 4 ... Wednesday
- 5 ... Thursday
- 6 ... Friday
- 7 ... Saturday

14-16 3 [WEIGHT] Weight of Record

Each record is assigned a weight based on the sampling rate of the place of marriage. This field is used to inflate tabular totals to State and MRA figures:

- 001 ... 100 percent
- 002 ... 50 percent
- 005 ... 20 percent
- 010 ... 10 percent
- 020 ... 5 percent

...

Fixed format data with a self-contained command

Quick and dirty

Syntax: `infix specification using filename`

where *specification* is

`[datatype] varname #[-#] [[datatype] varname #[-#]...]`

datatype is most relevant when the variable is a string and should be `str` when that is the case

```
. infix datayear 1-4 regnocc 5 divnocc 6
stateocc 7-8 marmon 9-10 marday 11-12
weekday 13 weight 14-16 using cpmarr.dat
(1357710 observations read)
```

```
. tab regnocc
```

regnocc	Freq.	Percent	Cum.
0	43,690	3.22	3.22
1	282,136	20.78	24.00
2	342,190	25.20	49.20
3	419,735	30.91	80.12
4	269,959	19.88	100.00
Total	1,357,710	100.00	

Fixed format data with a dictionary

Better in terms of readability and error minimization

Syntax: `infix using dfilename,`
`using(filename)`

where *dfilename* is the name of a dictionary file
filename is the name of the ASCII data file

Dictionary files for -infix-

-----start of cpmarr2.dct-----

```
infix dictionary      {  
    datayear          1-4  
    regnocc           5  
    divnocc           6  
    stateocc          7-8  
    marmon            9-10  
    marday            11-12  
    weekday           13  
    weight            14-16  
    str gnummar       24
```

* gnummar has non-numeric

* missing value codes

}

-----end of cpmarr2.dct-----

```
. infix using cpmarr2.dct, using(cpmarr.dat)
infix dictionary {
```

```
    datayear          1-4
```

```
    regnocc           5-5
```

```
    . . .
```

```
    bager24           64-64
```

```
    agedfr2           65-65
```

```
    agedfdt           66-67
```

```
}
```

```
(1357710 observations read)
```

```
. tab weekday [fw=weight]
```

weekday	Freq.	Percent	Cum.
1	948,982	7.39	7.39
2	646,429	5.03	12.43
3	615,630	4.79	17.22
4	633,377	4.93	22.15
5	753,934	5.87	28.03
6	2,288,807	17.83	45.85
7	6,952,020	54.15	100.00
Total	12,839,179	100.00	

Comma-delimited data

If your data are not in fixed format, they may be in comma-delimited format:

```
"Name", "STID", "Class", "HW1", "HW2", "HW4", "HW5", "HW6", "HW7", "HW8", "HW10", "HW11", "HW12", "HW13", "HW14", "HW15"  
1,159342,2,5,5,5,5,5,5,5,5,5,5,5,5,5  
2,90399,2,,,5,5,5,,,,,,,,,  
3,317693,2,5,5,5,5,5,5,5,5,5,,,  
4,317574,2,,,5,5,5,5,5,5,5,5,5,5,5  
5,188628,2,5,5,,5,5,5,5,5,5,5,5,5,  
6,207328,2,5,5,5,5,5,5,5,5,5,5,5,5,5  
7,445184,2,5,5,5,5,5,5,5,5,,5,5,,  
8,348699,2,5,5,5,5,5,,5,5,5,5,,,  
9,137397,2,5,5,5,5,5,5,5,5,5,5,5,5,5  
10,445873,2,5,5,5,5,,5,5,,5,5,,5,5  
11,463577,2,,,,,,,,,,,,,  
12,317905,2,5,5,5,5,5,5,5,5,5,5,5,5,5  
13,282425,2,5,5,5,5,5,5,5,5,5,,,,  
14,347640,2,5,5,5,5,5,5,5,5,5,5,5,5,5  
15,109834,2,5,5,5,5,5,,5,5,5,5,5,5,5  
16,315019,2,,,,,,,,,,,,,  
17,348322,2,5,5,5,5,5,5,5,5,5,5,5,5,5  
18,391099,2,5,5,5,5,5,5,5,5,5,5,5,,  
19,338614,2,5,5,5,5,5,5,5,5,5,5,5,5,5
```

This is an Excel spreadsheet saved in “CSV” format

Use the command `-insheet-`

Syntax: `insheet [varlist] using filename`

The first line of our ASCII file is interpreted by Stata as a set of variable names

In ASCII files without such names, you can optionally specify a *varlist*

Stata will detect whether each variable is numeric or a string

```
. insheet using "Gradebook Spring '05.csv"  
(16 vars, 57 obs)
```

```
. tab hw1, missing
```

HW1	Freq.	Percent	Cum.
5	47	82.46	82.46
.	10	17.54	100.00
Total	57	100.00	

General points

It will often take several tries to get the data successfully read in

Always check to make sure that the data were read in correctly

Check that all the values are reasonable

Check that values are logically consistent

Do this now to reduce the likelihood that you'll spend time analyzing bad data