import delimited — Import delimited text data

SyntaxMenuOptions for import delimitedOptions for export delimitedAlso see

Description Remarks and examples

Syntax

Load a delimited text file

import delimited [using] filename [, import_delimited_options]

Rename specified variables from a delimited text file

import delimited extvarlist using filename [, import_delimited_options]

Save data in memory to a delimited text file

export delimited [using] filename [if] [in] [, export_delimited_options]

Save subset of variables in memory to a delimited text file

export <u>delim</u>ited [varlist] using filename [if] [in] [, export_delimited_options]

import_delimited_options	Description
<pre>delimiters("chars"[, collapse asstring])</pre>	use chars as delimiters
<pre>rowrange([start][:end])</pre>	row range of data to load
<pre>colrange([start][:end])</pre>	column range of data to load
$\underline{varn}ames(\# nonames)$	treat row # of data as variable names or the data do not have variable names
case(preserve lower upper)	preserve the case or read variable names as lowercase (the default) or uppercase
<u>asdoub</u> le	import all floating-point data as doubles
<u>asflo</u> at	import all floating-point data as floats
clear	replace data in memory
bindquotes(loose strict nobind)	specify how to handle double quotes in data
$\overline{\texttt{strip}}$ quotes(yes no default)	remove or keep double quotes in data
<u>numericc</u> ols(<i>numlist</i> _all)	force specified columns to be numeric
<pre>stringcols(numlist _all)</pre>	force specified columns to be string
<pre>charset("charset")</pre>	set the character set used for importing ASCII text

extvarlist specifies variable names of imported columns.

export_delimited_options	Description
Main	
<u>delim</u> iter(" <i>char</i> " tab)	use char as delimiter
novarnames	do not write variable names on the first line
nolabel	output numeric values (not labels) of labeled variables
quote	always enclose strings in double quotes
 replace	overwrite existing <i>filename</i>

Menu

import delimited

File > Import > Text data (delimited, *.csv, ...)

export delimited

File > Export > Text data (delimited, *.csv, ...)

Description

import delimited reads into memory a text-delimited file from disk. Regardless of the program that created the file, import delimited reads text (ASCII) files in which there is one observation per line and the values are separated by commas, tabs, or some other delimiter. The first line of the file can contain the variable names. import delimited reads your data if you type

. import delimited *filename*

Stata has other commands for importing data. If you are not sure that import delimited will do what you are looking for, see [D] import and [U] 21 Entering and importing data.

export delimited, by default, writes data into a file in comma-separated (.csv) format. export delimited also allows you to specify any separation character delimiter that you prefer.

If *filename* is specified without an extension, .csv is assumed for both import delimited and export delimited. If *filename* contains embedded spaces, enclose it in double quotes.

Options for import delimited

delimiters("chars"[, collapse|asstring]) allows you to specify other separation characters. For instance, if values in the file are separated by a semicolon, specify delimiters(";"). By default, import delimited will check if the file is delimited by tabs or commas based on the first line of data. Specify delimiters("\t") to use a tab character, or specify delimiters("whitespace") to use whitespace as a delimiter.

collapse forces import delimited to treat multiple consecutive delimiters as just one delimiter.

- asstring forces import delimited to treat *chars* as one delimiter. By default, each character in *chars* is treated as an individual delimiter.
- rowrange([start]]:end]) specifies a range of rows within the data to load. start and end are integer row numbers.

- colrange([*start*][:*end*]) specifies a range of variables within the data to load. *start* and *end* are integer column numbers.
- varnames(#|nonames) specifies where or whether variable names are in the data. By default, import delimited tries to determine whether the file includes variable names. import delimited translates the names in the file to valid Stata variable names. The original names from the file are stored unmodified as variable labels.
 - varnames (#) specifies that the variable names are in row # of the data; any data before row # should not be imported.

varnames(nonames) specifies that the variable names are not in the data.

- case(preserve | lower | upper) specifies the case of the variable names after import. The default
 is case(lowercase).
- asdouble imports floating-point data as type double. The default storage type of the imported variables is determined by set type.
- asfloat imports floating-point data as type float. The default storage type of the imported variables is determined by set type.
- clear specifies that it is okay to replace the data in memory, even though the current data have not been saved to disk.
- bindquotes (loose | strict | nobind) specifies how import delimited handles double quotes in data. Specifying loose (the default) tells import delimited that it must have a matching open and closed double quote on the same line of data. strict tells import delimited that once it finds one double quote on a line of data, it should keep searching through the data for the matching double quote even if that double quote is on another line. Specifying nobind tells import delimited to ignore double quotes for binding.
- stripquotes(yes | no | default) tells import delimited how to handle double quotes. yes
 causes all double quotes to be stripped. no leaves double quotes in the data unchanged. default
 automatically strips quotes that can be identified as binding quotes. default also will identify
 two adjacent double quotes as a single double quote because some software encodes double quotes
 that way.
- numericcols(*numlist* | _all) forces the data type of the column numbers in *numlist* to be numeric. Specifying _all will import all data as numeric.
- stringcols(numlist | _all) forces the data type of the column numbers in numlist to be string. Specifying _all will import all data as strings.
- charset("charset") sets the character set used for importing ASCII text. Because the number of characters in natural languages far exceeds the number of printable character codes in ASCII, character sets allow more characters to be represented in ASCII to accommodate different languages. The default is charset("latin1") (ISO-8859-1 encoding). Specify charset("mac") for files containing Mac OS Roman encoding. Currently, only these two character sets are supported.

Options for export delimited

delimiter("char" | tab) allows you to specify other separation characters. For instance, if you want the values in the file to be separated by a semicolon, specify delimiter(";"). The default delimiter is a comma.

delimiter(tab) specifies that a tab character be used as the delimiter.

novarnames specifies that variable names not be written in the first line of the file; the file is to contain data values only.

nolabel specifies that the numeric values of labeled variables be written into the file rather than the label associated with each value.

quote specifies that string variables always be enclosed in double quotes. The default is to only double quote strings that contain spaces or the delimiter.

replace specifies that *filename* be replaced if it already exists.

Remarks and examples

stata.com

Remarks are presented under the following headings:

import delimited export delimited

import delimited

import delimited reads in text data where each data point is separated by a delimiter character. The two most common types of text data to import are comma-space-value (.csv) text files and tab-separated text files. import delimited can automatically detect either a comma or a tab as the delimiter. To import your data, type

. import delimited *filename*

import delimited reads your data if

- 1. it can find the file; and
- 2. the file meets import delimited's expectations as to its format.

If you type import delimited myfile, myfile.csv is read into Stata. If your file is called myfile.txt, type import delimited using myfile.txt. If typing import delimited *filename* does not produce the desired result, you may need to specify an option or try one of Stata's other import commands; see [D] import.

Example 1

We have a .csv data file on automobiles called auto.csv.

```
. copy http://www.stata.com/examples/auto.csv auto.csv
. type auto.csv
make,price,mpg,rep78,foreign
"AMC Concord",4099,22,3,"Domestic"
"AMC Pacer",4749,17,3,"Domestic"
"AMC Spirit",3799,22,,"Domestic"
"Buick Century",4816,20,3,"Domestic"
"Buick Century",4816,20,3,"Domestic"
"Buick Electra",7827,15,4,"Domestic"
"Buick LeSabre",5788,18,3,"Domestic"
"Buick Opel",4453,26,"Domestic"
"Buick Regal",5189,20,3,"Domestic"
"Buick Riviera",10372,16,3,"Domestic"
"Buick Skylark",4082,19,3,"Domestic"
```

This file was saved by a spreadsheet and can be read by typing

. import delimited auto

To look at what we just loaded, type

	make	price	mpg	rep78	foreign
1.	AMC Concord	4099	22	3	Domestic
2.	AMC Pacer	4749	17	3	Domestic
з.	AMC Spirit	3799	22		Domestic
4.	Buick Century	4816	20	3	Domestic
5.	Buick Electra	7827	15	4	Domestic
6.	Buick LeSabre	5788	18	3	Domestic
7.	Buick Opel	4453	26		Domestic
8.	Buick Regal	5189	20	3	Domestic
9.	Buick Riviera	10372	16	3	Domestic
10.	Buick Skylark	4082	19	3	Domestic

These data contain a combination of string and numeric variables. import delimited will determine the correct data type for each variable. You can also force the data type of a variable by using the numericcols() or stringcols() option.

▷ Example 2

import delimited allows you to read in a subset of the text data by using the rowrange() and colrange() options. To read rows 2 through 5 of auto.csv, you need to specify rowrange(3:6) because the first row of data contains the variable names.

```
. clear
. import delimited auto, rowrange(3:6)
(5 vars, 4 obs)
. list
```

	make	price	mpg	rep78	foreign
1.	AMC Pacer	4749	17	3	Domestic
2.	AMC Spirit	3799	22		Domestic
з.	Buick Century	4816	20	3	Domestic
4.	Buick Electra	7827	15	4	Domestic

We used rowrange(3:6) instead of rowrange(2:5) because row 1 of the data contained the variable names.

. list

To import the first three columns and last four rows of auto.csv, type

Buick Opel	4453	26
Buick Regal	5189	20
Buick Riviera	10372	16
Buick Skylark	4082	19
	Buick Regal	Buick Regal 5189 Buick Riviera 10372

Example 3

import delimited can handle delimiters other than commas and tabs. Suppose that you had the auto.txt file.

. type auto.txt	, showt	abs		
"AMC Concord"	4099	22	3	"Domestic"
"AMC Pacer"	4749	17	3	"Domestic"
"AMC Spirit"	3799	22	NA	"Domestic"
"Buick Century"	4816	20	3	"Domestic"
"Buick Electra"	7827	15	4	"Domestic"
"Buick LeSabre"	5788	18	3	"Domestic"
"Buick Opel"	4453	26	NA	"Domestic"
"Buick Regal"	5189	20	3	"Domestic"
"Buick Riviera"	10372	16	3	"Domestic"
"Buick Skylark"	4082	19	3	"Domestic"

We specified type's showtabs option so that no tabs are shown. These data are not tab-delimited or comma-delimited. If you use import delimited without any options, you will not get the results you expect.

```
. clear
. import delimited auto.txt
(1 var, 10 obs)
```

When import delimited tries to read data that have no tabs or commas, it is fooled into thinking that the data contain just one variable. You can use the delimiter() option to import the data correctly. delimiter(" ") tells import delimited to use spaces (" ") as the delimiter, and delimiter(, collapse) will treat multiple consecutive space delimiters as one delimiter.

4

. clear					
. import delin (5 vars, 10 ol		o.txt, deli	imiter(" "	, collapse)	
. describe					
Contains data					
obs:	10				
vars:	5				
size:	260				
	storage	display	value		
variable name	type	format	label	variable label	
make	str13	%13s			
price	int	%8.Og			
mpg	byte	%8.0g			
rep78	str2	%9s			
foreign	str8	%9s			

Sorted by:

Note: dataset has changed since last saved

. list

	make	price	mpg	rep78	foreign
1.	AMC Concord	4099	22	3	Domestic
2.	AMC Pacer	4749	17	3	Domestic
з.	AMC Spirit	3799	22	NA	Domestic
4.	Buick Century	4816	20	3	Domestic
5.	Buick Electra	7827	15	4	Domestic
6.	Buick LeSabre	5788	18	3	Domestic
7.	Buick Opel	4453	26	NA	Domestic
8.	Buick Regal	5189	20	3	Domestic
9.	Buick Riviera	10372	16	3	Domestic
10.	Buick Skylark	4082	19	3	Domestic

The data that were loaded now contain the correct number of variables and observations. However, the variable rep78 should be a numeric variable, but it was imported as a string because the value NA was used for missing values. To force rep78 to have a numeric storage type, use the option numericcols().

```
. clear
. import delimited auto.txt, delim(" ", collapse) numericcols(4)
(5 vars, 10 obs)
. describe
Contains data
  obs:
                   10
 vars:
                    5
 size:
                  250
               storage
                          display
                                      value
variable name
                          format
                                      label
                                                 variable label
                 type
                          %13s
make
                 str13
price
                 int
                          %8.0g
                          %8.0g
mpg
                 byte
                          %8.0g
rep78
                 byte
foreign
                 str8
                          %9s
```

Sorted by:

Note: dataset has changed since last saved

. list

	make	price	mpg	rep78	foreign
1.	AMC Concord	4099	22	3	Domestic
2.	AMC Pacer	4749	17	3	Domestic
3.	AMC Spirit	3799	22		Domestic
4.	Buick Century	4816	20	3	Domestic
5.	Buick Electra	7827	15	4	Domestic
6.	Buick LeSabre	5788	18	3	Domestic
7.	Buick Opel	4453	26		Domestic
8.	Buick Regal	5189	20	3	Domestic
9.	Buick Riviera	10372	16	3	Domestic
10.	Buick Skylark	4082	19	3	Domestic

export delimited

export delimited creates a comma-separated text file from the Stata dataset in memory. If your goal is to send data to another Stata user, you could use export delimited, but it is better to send a .dta dataset. This will work even if you use Stata for Windows and your colleague uses Stata for Mac or Unix. All versions of Stata can read each other's .dta files.

To view other methods for moving your data into other applications, see [D] export.

Example 4

To save the data currently in memory into a specified .csv file, type

```
. use http://www.stata-press.com/data/r13/auto, clear
(1978 Automobile Data)
. export delimited myauto
file myauto.csv saved
```

4

▷ Example 5

You can also export a subset of the data in memory by typing

```
. use http://www.stata-press.com/data/r13/auto
(1978 Automobile Data)
. export delimited make mpg rep78 foreign in 1/10 using myauto
file myauto.csv already exists
r(602);
```

If the file already exists, you can use replace to write over it:

```
. export delimited make mpg rep78 foreign in 1/10 using myauto, replace
. type myauto.csv
make,mpg,rep78,foreign
AMC Concord,22,3,Domestic
AMC Pacer,17,3,Domestic
AMC Spirit,22,,Domestic
Buick Century,20,3,Domestic
Buick Electra,15,4,Domestic
Buick LeSabre,18,3,Domestic
Buick LeSabre,18,3,Domestic
Buick Regal,20,3,Domestic
Buick Riviera,16,3,Domestic
Buick Riviera,16,3,Domestic
```

Also see

- [D] export Overview of exporting data from Stata
- [D] import Overview of importing data into Stata