How do I use file read to read in information in an external text file?

Authored by

stats writer

July 1, 2024

Using file read to read in information from an external text file is a simple and efficient way to access data stored in a separate file. This process involves opening the desired file, reading its contents, and then closing the file once the necessary information has been retrieved. By using this method, one can easily extract data from an external source and use it in their program or application. It is a common practice used in various programming languages and is particularly useful when dealing with large amounts of data. Overall, utilizing file read to read in information from an external text file is a convenient and effective way to incorporate external data into your code.

How do I use file read to read in information in an external text file? | Stata FAQ

The file read command can be used to read information from an external text file. Note that if you are trying to read in a dataset in ascii format, another command, such as infile or insheet is probably more appropriate. If you have a list of some sort in a text file, or some other type of information, which is not formatted the way a dataset would be, or that you do not want to treat the way you would treat a dataset, then it is appropriate to use file read.

Our example reads in information from a text file, and displays it in Stata. We start with a file named

https://stats.idre.ucla.edu/wp-content/uploads/2016/02/example.txt (click here to download: https://stats.idre.ucla.edu/wp-content/uploads/2016/02/example.txt), it has seven lines of text that we want to display in Stata. If we open the text file, it looks like this:

This is the first line
This is the second line
The third line
The fourth line
The fifth line
The sixth line
The last line (7)

Stata has a group of commands all of which start with the command file,
these commands allow us to read and/or write information from text files. The structure of these commands is similar. As mentioned previously, they all start
with the command file, followed by the type of operation we want to perform (in this case, open the file). The operation (e.g.
open) is followed by a name that is used to identify the

text file (known as a handle).
All of this is followed by something else, depending on what we are doing with the file.
Each open file has a name associated with it, (in our example the
file is called myfile). It is necessary to assign each file a name because it is possible to have more than
one file open at a time, meaning that we need some way to tell Stata which file
we are referring to with any given command.

The first line below
instructs Stata to open a file (file open), and assigns it the name myfile. The word using followed by a file path and
name tells Stata where the text file we want to read is located. In the options
(i.e. following the comma) we
specify read which tells Stata that we are going to be reading (i.e.
getting information from, but not adding information to) the file. The next line of syntax reads the first line of the file we have named
myfile.

When Stata "reads" a line from a file it reads a single line and stops (unless we tell it to do otherwise). The information on that line is placed in a local macro which the user can then access. Next time we tell Stata to read from that file, it will read the next line, and so forth working it's way through the file. The second line of code below tells Stata to read a line from the file we have called myfile by starting with the command (i.e. file), then the action we want to perform on the file (i.e. read), followed by the name of the file (i.e. myfile), and finally, by giving Stata the name of the local macro into which we want to place the information read from the file (i.e. line).

```
file open myfile using "d:filelist.txt", read
file read myfile line
```

Before we proceed there are a few more things you need to know in order for the code

below to make sense. We know that when we use the file read command, Stata reads lines one a time, and that each time we ask it to read a line, it will read the next line in the file. What does Stata do when it reaches the end of the file? As we've just said, each time Stata reads a line from a file, it returns that line in the local macro we specify (in this example line), but it also returns something else, a returned result called "r(eof)". (See our Stata FAQ How can I access information stored after I run a command in Stata (returned results)? for more information on returned results.) As long as we are not at the end of the file, r(eof) is equal to zero, when the end of the file is reached, Stata sets r(eof) equal to one, and the local macro specified in the command will be empty.

In the lines of code above, we read the first line in myfile, so now we have the information in the first line of myfile in the

local macro line, and r(eof) is equal to zero. In the first line
of syntax below, the command while tells Stata to repeat whatever is
inside the curly braces for as long as the statement following it is true. In
this case the statement is r(eof)==0, so, as long as r(eof) is equal to
zero, that is, as long as we are not at the end of the file, Stata should keep
repeating whatever is in the curly brackets. The
next line of syntax uses the display command to show the user the
contents of the local macro line. Stata uses a ` (it's below the ~ character at
the top of your keyboard) followed by the name of the macro, and then a ' (a
normal apostrophe) to represent whatever you want to fill in, since we assigned
the name "line" to the line of information from myfile, we type
`line'. Stata replaces the `line' with whatever is stored in the
local macro line, so that when it interprets the first line

**below, what it
actually sees is: display "This is the first line" (note that the double
quotation marks, i.e. " and " are necessary). After we display the line of text, the third line of syntax instructs Stata to read
the next line in myfile. The final line of syntax closes the while loop
using a curly
bracket (}).**

```
while r(eof)==0 {
display "`line'"
file read myfile line
}
```

**When you run the above code (you will need to use a .do file to do this), you will see the output below in your Stata output window.
The first four lines are Stata echoing back the while command, as you entered
it. The next seven lines are the output from the program, that is, the contents**

of example.txt, displayed one line at a time.

```
while r(eof)==0 {
2. display "`line'"
3. file read myfile line
4. }
This is the first line
This is the second line
The third line
The fourth line
The fifth line
The sixth line
The last line (7)
```

The final thing we need to do is close the file. This is not strictly necessary in order to get Stata to display the contents of the file, but, it is good practice. The line below tells Stata that we would like to close the file known as myfile. This means we will not be able to read or write to this file unless we open it again.

```
file close myfile
```

The above example is a fairly basic example of what can be done with file read.
File read can be used in conjunction with other Stata commands to accomplish a
variety of tasks. For example, if you wanted to take only the first (or third,
etc.) word from each line, you could use the string function word(s,n)
to display only the first word of each line, the code below does this. The only
change that is necessary is to the line beginning with the display
command.

```
file open myfile using "D:example.txt", read
file read myfile line

while r(eof)==0 {
display "`=word("`line'",1)'"
file read myfile line
}

file close myfile
```

For another more advanced use of the file read

command, see the FAQ page

How can I combine a large number of files?.

See also