Title

stata.com

byteorder( ) — Byte order used by computer

Description

`byteorder()` returns 1 if the computer is HILO (records most significant byte first) and returns 2 if LOHI (records least significant byte first).

Syntax

real scalar `byteorder()`

Remarks and examples

stata.com

Pretend that the values 00 and 01 are recorded at memory positions 58 and 59 and that you know what is recorded there is a 2-byte integer. How should the 2-byte number be interpreted: as 0001 (meaning 1) or 0100 (meaning 256 in decimal)? For different computers, the answer varies. For HILO computers, the number is to be interpreted as 0001. For LOHI computers, the number is interpreted as 0100.

Regardless, it does not matter because the computer is consistent with itself. An issue arises, however, when we write binary files that may be read on computers using a different byte order or when we read files from computers that used a different byte order.

Stata and Mata automatically handle these problems for you, so you may wish to stop reading. `byteorder()`, however, is the tool on which the solution is based. If you intend to write code based on your own binary-file format or to write code to process the binary files of others, then you may need to use it.

There are two popular solutions to the byte-order problem: (1) write the file in a known byte order or (2) write the file by using whichever byte order is convenient and record the byte order used in the file. StataCorp tends to use the second, but others who have worried about this problem have used both solutions.

In solution (1), it is usually agreed that the file will be written in HILO order. If you are using a HILO computer, you write and read files the ordinary way, ignoring the problem altogether. If you are on a LOHI computer, however, you must reverse the bytes before placing them in the file. If you are writing code designed to execute on both kinds of computers, you must write code for both circumstances, and you must consider the problem when both reading and writing the files.

In solution (2), files are written LOHI or HILO, depending solely on the computer being used. Early in the file, however, the byte order is recorded. When reading the file, you compare the order in which the file is recorded with the order of the computer and, if they are different, you reverse bytes.

Mata-buffered I/O utilities will automatically reverse bytes for you. See [M-5] **bufio( )**.

Conformability

`byteorder()`:

result: 1×1

Diagnostics

None.

Also see

[M-5] **bufio( )** — Buffered (binary) I/O

[M-5] **stataversion( )** — Version of Stata being used

[M-4] **programming** — Programming functions