

**substr()** — Extract substring

Description  
Also see

Syntax

Remarks and examples

Conformability

Diagnostics

## Description

`substr(s, b, l)` returns the substring of ASCII string *s* starting at position *b* and continuing for a length of *l* characters.

For non-ASCII strings, *b* and *l* are interpreted as byte positions. To obtain character-based substrings of Unicode strings, see [M-5] [usubstr\(\)](#).

`substr(s, b)` is equivalent to `substr(s, b, .)` for strings that do not contain binary 0. If there is a binary 0 to the right of *b*, the substring from *b* up to but not including the binary 0 is returned.

When arguments are not scalar, `substr()` returns element-by-element results.

## Syntax

*string matrix*    `substr(string matrix s, real matrix b, real matrix l)`

*string matrix*    `substr(string matrix s, real matrix b)`

## Remarks and examples

stata.com

`substr(s, b, l)` returns the substring of ASCII string *s* starting at position *b* and continuing for a length of *l*, where

1. *b* specifies the starting position; the first character of the string is  $b = 1$ .
2.  $b > 0$  is interpreted as distance from the start of the string;  $b = 2$  means starting at the second character.
3.  $b < 0$  is interpreted as distance from the end of string;  $b = -1$  means starting at the last character;  $b = -2$  means starting at the second from the last character.
4. *l* specifies the length;  $l = 2$  means for two characters.
5.  $l < 0$  is treated the same as  $l = 0$ : no characters are copied.
6.  $l \geq .$  is interpreted to mean to the end of the string.

`substr(s, b)` is equivalent to `substr(s, b, .)` for strings that do not contain binary 0. If there is a binary 0 to the right of *b*, the substring from *b* up to but not including the binary 0 is returned.

If your string contains Unicode characters, see [M-5] [usubstr\(\)](#) and [M-5] [udsubstr\(\)](#).

## Conformability

`substr(s, b, l)`:

*s*:  $r_1 \times c_1$

*b*:  $r_2 \times c_2$

*l*:  $r_3 \times c_3$ ; *s*, *b*, and *l* r-conformable

*result*:  $\max(r_1, r_2, r_3) \times \max(c_1, c_2, c_3)$

`substr(s, b)`:

*s*:  $r_1 \times c_1$

*b*:  $r_2 \times c_2$ ; *s* and *b* r-conformable

*result*:  $\max(r_1, r_2) \times \max(c_1, c_2)$

## Diagnostics

In `substr(s, b, l)` and `substr(s, b)`, if *b* describes a position before the beginning of the string or after the end, "" is returned. If *b* + *l* describes a position to the right of the end of the string, results are as if a smaller value for *l* were specified.

## Also see

[M-5] [subinstr\(\)](#) — Substitute text

[M-5] [\\_substr\(\)](#) — Substitute into string

[M-5] [usubinstr\(\)](#) — Replace Unicode substring

[M-5] [usubstr\(\)](#) — Extract Unicode substring

[M-5] [\\_usubstr\(\)](#) — Substitute into Unicode string

[M-4] [String](#) — String manipulation functions

[U] [12.4.2 Handling Unicode strings](#)

Stata, Stata Press, and Mata are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow and NetCourseNow are trademarks of StataCorp LLC. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2023 StataCorp LLC, College Station, TX, USA. All rights reserved.



For suggested citations, see the FAQ on [citing Stata documentation](#).