|  |  |
| --- | --- |
| curve {graphics} | R Documentation |

**Draw Function Plots**

**Description**

Draws a curve corresponding to a function over the interval [from, to]. curve can plot also an expression in the variable xname, default x.

**Usage**

curve(expr, from = NULL, to = NULL, n = 101, add = FALSE,

type = "l", xname = "x", xlab = xname, ylab = NULL,

log = NULL, xlim = NULL, ...)

## S3 method for class 'function'

plot(x, y = 0, to = 1, from = y, xlim = NULL, ylab = NULL, ...)

**Arguments**

|  |  |
| --- | --- |
| expr | The name of a function, or a [call](http://127.0.0.1:14695/library/graphics/help/call) or an [expression](http://127.0.0.1:14695/library/graphics/help/expression) written as a function of x which will evaluate to an object of the same length as x. |
| x | a ‘vectorizing’ numeric **R** function. |
| y | alias for from for compatibility with plot |
| from, to | the range over which the function will be plotted. |
| n | integer; the number of x values at which to evaluate. |
| add | logical; if TRUE add to an already existing plot; if NA start a new plot taking the defaults for the limits and log-scaling of the x-axis from the previous plot. Taken as FALSE (with a warning if a different value is supplied) if no graphics device is open. |
| xlim | NULL or a numeric vector of length 2; if non-NULL it provides the defaults for c(from, to) and, unless add = TRUE, selects the x-limits of the plot – see [plot.window](http://127.0.0.1:14695/library/graphics/help/plot.window). |
| type | plot type: see [plot.default](http://127.0.0.1:14695/library/graphics/help/plot.default). |
| xname | character string giving the name to be used for the x axis. |
| xlab, ylab, log, ... | labels and [graphical parameters](http://127.0.0.1:14695/library/graphics/help/graphical%20parameters) can also be specified as arguments. See ‘Details’ for the interpretation of the default for log.  For the "function" method of plot, ... can include any of the other arguments of curve, except expr. |

**Details**

The function or expression expr (for curve) or function x (for plot) is evaluated at n points equally spaced over the range [from, to]. The points determined in this way are then plotted.

If either from or to is NULL, it defaults to the corresponding element of xlim if that is not NULL.

What happens when neither from/to nor xlim specifies both x-limits is a complex story. For plot(<function>) and for curve(add = FALSE) the defaults are *(0, 1)*. For curve(add = NA) and curve(add = TRUE) the defaults are taken from the x-limits used for the previous plot. (This differs from versions of **R** prior to 2.14.0.)

The value of log is used both to specify the plot axes (unless add = TRUE) and how ‘equally spaced’ is interpreted: if the x component indicates log-scaling, the points at which the expression or function is plotted are equally spaced on log scale.

The default value of log is taken from the current plot when add = TRUE, whereas if add = NA the x component is taken from the existing plot (if any) and the y component defaults to linear. For add = FALSE the default is ""

This used to be a quick hack which now seems to serve a useful purpose, but can give bad results for functions which are not smooth.

For expensive-to-compute expressions, you should use smarter tools.

The way curve handles expr has caused confusion. It first looks to see if expr is a [name](http://127.0.0.1:14695/library/graphics/help/name) (also known as a symbol), in which case it is taken to be the name of a function, and expr is replaced by a call to expr with a single argument with name given by xname. Otherwise it checks that expr is either a [call](http://127.0.0.1:14695/library/graphics/help/call) or an [expression](http://127.0.0.1:14695/library/graphics/help/expression), and that it contains a reference to the variable given by xname (using [all.vars](http://127.0.0.1:14695/library/graphics/help/all.vars)): anything else is an error. Then expr is evaluated in an environment which supplies a vector of name given by xname of length n, and should evaluate to an object of length n. Note that this means that curve(x, ...) is taken as a request to plot a function named x (and it is used as such in the function method for plot).

The plot method can be called directly as plot.function.

**Value**

A list with components x and y of the points that were drawn is returned invisibly.

**Warning**

For historical reasons, add is allowed as an argument to the "function" method of plot, but its behaviour may surprise you. It is recommended to use add only with curve.

**See Also**

[splinefun](http://127.0.0.1:14695/library/graphics/help/splinefun) for spline interpolation, [lines](http://127.0.0.1:14695/library/graphics/help/lines).

**Examples**

plot(qnorm) # default range c(0, 1) is appropriate here,

# but end values are -/+Inf and so are omitted.

plot(qlogis, main = "The Inverse Logit : qlogis()")

abline(h = 0, v = 0:2/2, lty = 3, col = "gray")

curve(sin, -2\*pi, 2\*pi, xname = "t")

curve(tan, xname = "t", add = NA,

main = "curve(tan) --> same x-scale as previous plot")

op <- par(mfrow = c(2, 2))

curve(x^3 - 3\*x, -2, 2)

curve(x^2 - 2, add = TRUE, col = "violet")

## simple and advanced versions, quite similar:

plot(cos, -pi, 3\*pi)

curve(cos, xlim = c(-pi, 3\*pi), n = 1001, col = "blue", add = TRUE)

chippy <- function(x) sin(cos(x)\*exp(-x/2))

curve(chippy, -8, 7, n = 2001)

plot (chippy, -8, -5)

for(ll in c("", "x", "y", "xy"))

curve(log(1+x), 1, 100, log = ll, sub = paste0("log = '", ll, "'"))

par(op)