|  |  |
| --- | --- |
| lapply {base} | R Documentation |

**Apply a Function over a List or Vector**

**Description**

lapply returns a list of the same length as X, each element of which is the result of applying FUN to the corresponding element of X.

sapply is a user-friendly version and wrapper of lapply by default returning a vector, matrix or, if simplify = "array", an array if appropriate, by applying simplify2array(). sapply(x, f, simplify = FALSE, USE.NAMES = FALSE) is the same as lapply(x, f).

vapply is similar to sapply, but has a pre-specified type of return value, so it can be safer (and sometimes faster) to use.

replicate is a wrapper for the common use of sapply for repeated evaluation of an expression (which will usually involve random number generation).

simplify2array() is the utility called from sapply() when simplify is not false and is similarly called from [mapply](http://127.0.0.1:14695/library/base/help/mapply)().

**Usage**

lapply(X, FUN, ...)

sapply(X, FUN, ..., simplify = TRUE, USE.NAMES = TRUE)

vapply(X, FUN, FUN.VALUE, ..., USE.NAMES = TRUE)

replicate(n, expr, simplify = "array")

simplify2array(x, higher = TRUE)

**Arguments**

|  |  |
| --- | --- |
| X | a vector (atomic or list) or an [expression](http://127.0.0.1:14695/library/base/help/expression) object. Other objects (including classed objects) will be coerced by base::[as.list](http://127.0.0.1:14695/library/base/help/as.list). |
| FUN | the function to be applied to each element of X: see ‘Details’. In the case of functions like +, %\*%, the function name must be backquoted or quoted. |
| ... | optional arguments to FUN. |
| simplify | logical or character string; should the result be simplified to a vector, matrix or higher dimensional array if possible? For sapply it must be named and not abbreviated. The default value, TRUE, returns a vector or matrix if appropriate, whereas if simplify = "array" the result may be an [array](http://127.0.0.1:14695/library/base/help/array) of “rank” (*=*length(dim(.))) one higher than the result of FUN(X[[i]]). |
| USE.NAMES | logical; if TRUE and if X is character, use X as [names](http://127.0.0.1:14695/library/base/help/names) for the result unless it had names already. Since this argument follows ... its name cannot be abbreviated. |
| FUN.VALUE | a (generalized) vector; a template for the return value from FUN. See ‘Details’. |
| n | integer: the number of replications. |
| expr | the expression (a [language object](http://127.0.0.1:14695/library/base/help/language%20object), usually a call) to evaluate repeatedly. |
| x | a list, typically returned from lapply(). |
| higher | logical; if true, simplify2array() will produce a (“higher rank”) array when appropriate, whereas higher = FALSE would return a matrix (or vector) only. These two cases correspond to sapply(\*, simplify = "array") or simplify = TRUE, respectively. |

**Details**

FUN is found by a call to [match.fun](http://127.0.0.1:14695/library/base/help/match.fun) and typically is specified as a function or a symbol (e.g., a backquoted name) or a character string specifying a function to be searched for from the environment of the call to lapply.

Function FUN must be able to accept as input any of the elements of X. If the latter is an atomic vector, FUN will always be passed a length-one vector of the same type as X.

Arguments in ... cannot have the same name as any of the other arguments, and care may be needed to avoid partial matching to FUN. In general-purpose code it is good practice to name the first two arguments X and FUN if ... is passed through: this both avoids partial matching to FUN and ensures that a sensible error message is given if arguments named X or FUN are passed through ....

Simplification in sapply is only attempted if X has length greater than zero and if the return values from all elements of X are all of the same (positive) length. If the common length is one the result is a vector, and if greater than one is a matrix with a column corresponding to each element of X.

Simplification is always done in vapply. This function checks that all values of FUN are compatible with the FUN.VALUE, in that they must have the same length and type. (Types may be promoted to a higher type within the ordering logical < integer < double < complex, but not demoted.)

Users of S4 classes should pass a list to lapply and vapply: the internal coercion is done by the as.list in the base namespace and not one defined by a user (e.g., by setting S4 methods on the base function).

lapply and vapply are [primitive](http://127.0.0.1:14695/library/base/help/primitive) functions.

**Value**

For lapply, sapply(simplify = FALSE) and replicate(simplify = FALSE), a list.

For sapply(simplify = TRUE) and replicate(simplify = TRUE): if X has length zero or n = 0, an empty list. Otherwise an atomic vector or matrix or list of the same length as X (of length n for replicate). If simplification occurs, the output type is determined from the highest type of the return values in the hierarchy NULL < raw < logical < integer < double < complex < character < list < expression, after coercion of pairlists to lists.

vapply returns a vector or array of type matching the FUN.VALUE. If length(FUN.VALUE) == 1 a vector of the same length as X is returned, otherwise an array. If FUN.VALUE is not an [array](http://127.0.0.1:14695/library/base/help/array), the result is a matrix with length(FUN.VALUE) rows and length(X) columns, otherwise an array a with [dim](http://127.0.0.1:14695/library/base/help/dim)(a) == c(dim(FUN.VALUE), length(X)).

The (Dim)names of the array value are taken from the FUN.VALUE if it is named, otherwise from the result of the first function call. Column names of the matrix or more generally the names of the last dimension of the array value or names of the vector value are set from X as in sapply.

**Note**

sapply(\*, simplify = FALSE, USE.NAMES = FALSE) is equivalent to lapply(\*).

For historical reasons, the calls created by lapply are unevaluated, and code has been written (e.g., bquote) that relies on this. This means that the recorded call is always of the form FUN(X[[i]], ...), with i replaced by the current (integer or double) index. This is not normally a problem, but it can be if FUN uses [sys.call](http://127.0.0.1:14695/library/base/help/sys.call) or [match.call](http://127.0.0.1:14695/library/base/help/match.call) or if it is a primitive function that makes use of the call. This means that it is often safer to call primitive functions with a wrapper, so that e.g. lapply(ll, function(x) is.numeric(x)) is required to ensure that method dispatch for is.numeric occurs correctly.

If expr is a function call, be aware of assumptions about where it is evaluated, and in particular what ... might refer to. You can pass additional named arguments to a function call as additional named arguments to replicate: see ‘Examples’.

**References**

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) *The New S Language*. Wadsworth & Brooks/Cole.

**See Also**

[apply](http://127.0.0.1:14695/library/base/help/apply), [tapply](http://127.0.0.1:14695/library/base/help/tapply), [mapply](http://127.0.0.1:14695/library/base/help/mapply) for applying a function to **m**ultiple arguments, and [rapply](http://127.0.0.1:14695/library/base/help/rapply) for a **r**ecursive version of lapply(), [eapply](http://127.0.0.1:14695/library/base/help/eapply) for applying a function to each entry in an [environment](http://127.0.0.1:14695/library/base/help/environment).

**Examples**

require(stats); require(graphics)

x <- list(a = 1:10, beta = exp(-3:3), logic = c(TRUE,FALSE,FALSE,TRUE))

# compute the list mean for each list element

lapply(x, mean)

# median and quartiles for each list element

lapply(x, quantile, probs = 1:3/4)

sapply(x, quantile)

i39 <- sapply(3:9, seq) # list of vectors

sapply(i39, fivenum)

vapply(i39, fivenum,

c(Min. = 0, "1st Qu." = 0, Median = 0, "3rd Qu." = 0, Max. = 0))

## sapply(\*, "array") -- artificial example

(v <- structure(10\*(5:8), names = LETTERS[1:4]))

f2 <- function(x, y) outer(rep(x, length.out = 3), y)

(a2 <- sapply(v, f2, y = 2\*(1:5), simplify = "array"))

a.2 <- vapply(v, f2, outer(1:3, 1:5), y = 2\*(1:5))

stopifnot(dim(a2) == c(3,5,4), all.equal(a2, a.2),

identical(dimnames(a2), list(NULL,NULL,LETTERS[1:4])))

hist(replicate(100, mean(rexp(10))))

## use of replicate() with parameters:

foo <- function(x = 1, y = 2) c(x, y)

# does not work: bar <- function(n, ...) replicate(n, foo(...))

bar <- function(n, x) replicate(n, foo(x = x))

bar(5, x = 3)