stats (version 3.6.2)

integrate: Integration of One-Dimensional Functions

Description

Adaptive quadrature of functions of one variable over a finite or infinite interval.

Usage

```
integrate(f, lower, upper, …, subdivisions = 100L,
          rel.tol = .Machine$double.eps^0.25, abs.tol = rel.tol,
          stop.on.error = TRUE, keep.xy = FALSE, aux = NULL)
```

Arguments

`f`

an R function taking a numeric first argument and returning a numeric vector of the same length. Returning a non-finite element will generate an error.

`lower, upper`

the limits of integration. Can be infinite.

…

additional arguments to be passed to `` `f` ``.

`subdivisions`

the maximum number of subintervals.

`rel.tol`

relative accuracy requested.

`abs.tol`

absolute accuracy requested.

stop.on.error logical. If true (the default) an error stops the function. If false some errors will give a result with a warning in the `message` component.

keep.xy unused. For compatibility with S.

aux unused. For compatibility with S.

Value

A list of class `"integrate"` with components

value

the final estimate of the integral.

abs.error

estimate of the modulus of the absolute error.

subdivisions

the number of subintervals produced in the subdivision process.

message

`"OK"` or a character string giving the error message.

call

the matched call.

Details

Note that arguments after `…` must be matched exactly.

If one or both limits are infinite, the infinite range is mapped onto a finite interval.

For a finite interval, globally adaptive interval subdivision is used in connection with extrapolation by Wynn's Epsilon algorithm, with the basic step being Gauss--Kronrod quadrature.

quadrature.

`rel.tol` cannot be less than `max(50*.Machine$double.eps, 0.5e-28)` if `abs.tol <= 0`.

In R versions \(\le\) 3.2.x, the first entries of `lower` and `upper` were used whereas an error is signalled now if they are not of length one.

References

R. Piessens, E. deDoncker--Kapenga, C. Uberhuber, D. Kahaner (1983) *Quadpack: a Subroutine Package for Automatic Integration*; Springer Verlag.

Examples

```
# NOT RUN {
integrate(dnorm, -1.96, 1.96)
integrate(dnorm, -Inf, Inf)

## a slowly-convergent integral
integrand <- function(x) {1/((x+1)*sqrt(x))}
integrate(integrand, lower = 0, upper = Inf)

## don't do this if you really want the integral from 0 to Inf
integrate(integrand, lower = 0, upper = 10)
integrate(integrand, lower = 0, upper = 100000)
integrate(integrand, lower = 0, upper = 1000000, stop.on.error = FALSE)

## some functions do not handle vector input properly
f <- function(x) 2.0
try(integrate(f, 0, 1))
integrate(Vectorize(f), 0, 1)  ## correct
integrate(function(x) rep(2.0, length(x)), 0, 1)  ## correct

## integrate can fail if misused
integrate(dnorm, 0, 2)
integrate(dnorm, 0, 20)
integrate(dnorm, 0, 200)
integrate(dnorm, 0, 2000)
integrate(dnorm, 0, 20000) ## fails on many systems
integrate(dnorm, 0, Inf)   ## works
# }
# NOT RUN {
integrate(dnorm, 0:1, 20) #-> error!
## "silently" gave  integrate(dnorm, 0, 20)  in earlier versions of R
# }
```