

## Description

`pmledecon(pmledecon(ob,error,supp,n,lmd,R,sts,subid,conv))`

Given a sample with additive measurement error, `pmledecon` estimates the deconvolution density - that is, the density of the underlying distribution of the sample without measurement error. The method maximises the log-likelihood of the estimated density, plus a quadratic smoothness penalty. The distribution of the measurement error can be either a known family, or can be estimated from a "pure error" sample. For known error distributions, `pmledecon` supports Normal, Laplace or Beta distributed error. For unknown error distribution, a pure error sample independent from the data is used.

`ob` is the contaminated sample. `error` can either be a "pure error" sample, or a list consisting of the name of a distribution ("Normal", "Laplace" and "Beta" are currently supported) with the parameters. `supp` is a vector of values at which to compute the deconvolved density. Instead of a vector of support values, the parameter `n` sets the number of points in this vector, and the function sets these evenly spaced between the estimated endpoints. `lmd` is the smoothness penalty - larger values give more smooth estimated density functions. Alternatively, `R` can be used to set the value of `lmd` based on the data. Smaller values of `R` give smoother estimated densities. `sts` is the approximate step-size for numerical integration used to calculate the convolution. `subid` is a logical parameter indicating whether to output progress during the fitting. This can be useful for identifying problematic cases where the program fails to make progress and will repeatedly print the same number. If `conv=TRUE`, the function will return the mean convolved log-likelihood of the final solution.

Example:

```
> library(pmledecon)
> sz=esz=30
> set.seed(45217)
> truth=rnorm(sz,0,1)
> error=rnorm(esz,0,2)
> ob=truth+error
> error1=rnorm(esz,0,2)
> est=pmledecon(ob,error1)

[1] 1
[1] 2
[1] 3
[1] 4
[1] 5

> plot(density(ob,n=1000),col="red",lwd=2,lty=3,type="l",ylim=c(0,0.4),
+       xlab="",main="unknown error")
> lines(seq(-10,10,length.out=1000),dnorm(seq(-10,10,length.out=1000),0,1),
+       lwd=2,lty=4,col="green")
> lines(est$sup,est$f,lwd=2)
```

```
> legend("topright", lty=c(1,4,3),col=c("black","green","red"),lwd=2,  
+       legend=c("Pmle","true density","kernel density of data"))
```