Package ‘drmdel’

February 19, 2015

Version 1.3.1

Date 2014-12-31

Title Dual Empirical Likelihood Inference under Density Ratio Models in the Presence of Multiple Samples

Description Dual empirical likelihood (DEL) inference under semiparametric density ratio models (DRM) in the presence of multiple samples, including population cumulative distribution function estimation, quantile estimation and comparison, density estimation, composite hypothesis testing for DRM parameters which encompasses testing for changes in population distribution functions as a special case, etc.

Author Song Cai

Maintainer Song Cai <scai@math.carleton.ca>

Depends R (>= 2.11)

License GPL-3

NeedsCompilation yes

Repository CRAN

Date/Publication 2015-01-21 21:28:38

R topics documented:

cdfDRM *Estimate the CDF of the populations under the DRM*

Description

Suppose we have m+1 samples, labeled as 0, 1, ..., m, whose population distributions satisfy the density ratio model (DRM) (see `drmdel` for the definition of DRM). The `cdfDRM` function estimates the cumulative distribution functions (CDF), F_k, of the populations.

Usage

```
cdfDRM(k, x=NULL, drmfit, interpolation=TRUE)
```

Arguments

`k` a vector of labels of populations whose CDFs are to be estimated, with k[i] = 0, 1, ..., m.

`x` It can be:

(1) a list whose length is the same as the argument "k". The i^{th} component of this list must be a vector of values at which the CDF of population k[i] is estimated.

(2) a single vector of values, in which case, each CDF is estimated at the same values given by this vector.

(3) NULL (default), in which case, each CDF is estimated at the values of all the observed data points.

`drmfit` a fitted DRM object (an output from the `drmdel` function). See `drmdel` for details.

`interpolation` a logical variable specifying whether to linearly interpolate the estimated CDF to make a smooth CDF estimate; The default is TRUE.

Value

A list whose length is the same as the argument "k". The i^{th} component of the list is a data frame with the following two columns:

x: values at which the CDF of population k[i] is estimated.

cdf_est: the corresponding estimated CDF values of population k[i].

Examples

```
# Data generation
set.seed(25)
n_samples <- c(100, 200, 180, 150, 175)  # sample sizes
x0 <- rgamma(n_samples[1], shape=5, rate=1.8)
x1 <- rgamma(n_samples[2], shape=12, rate=1.2)
x2 <- rgamma(n_samples[3], shape=12, rate=1.2)
x3 <- rgamma(n_samples[4], shape=18, rate=5)
x4 <- rgamma(n_samples[5], shape=25, rate=2.6)
```

```
x <- c(x0, x1, x2, x3, x4)

# Fit a DRM with the basis function q(x) = (x, log(abs(x))),
# which is the basis function for gamma family. This basis
# function is the built-in basis function 6.
drmfit <- drmdel(x=x, n_samples=n_samples, basis_func=6)

# Estimate the CDF of population 1 at c(3, 7.5, 11) and that
# of population 3 at c(2, 6).
cdf_est1 <- cdfDRM(k=c(1, 3), x=list(c(3, 7.5, 11), c(2, 6)),
                   drmfit=drmfit)
# Show the output.
names(cdf_est1)
cdf_est1$F1
cdf_est1$F3

# Estimate the CDF of population 2 and 4 at the observed
# data points.
cdf_est <- cdfDRM(k=c(2, 4), drmfit=drmfit)
# Show the output.
names(cdf_est)
cdf_est$F2
cdf_est$F4
```

densityDRM *Estimate the density of the populations under the DRM*

Description

Suppose we have m+1 samples, labeled as $0, 1, \ldots, m$, whose population distributions satisfy the density ratio model (DRM) (see drmdel for the definition of DRM). The densityDRM function estimates the density, $dF_k(x)$, of the k$^{\text{th}}$, $k = 0, 1, \ldots, m$, population using a kernel density estimator with weights been the estimated $dF_k(x)$'s at the observed data points.

Usage

```
densityDRM(k, drmfit, interpolation=TRUE, ...)
```

Arguments

k	the label of the population whose density is to be estimated, k = 0, 1, ..., m. It must be a single value, not a vector.
drmfit	a fitted DRM object (an output from the drmdel function). See drmdel for details.
interpolation	a logical variable to be passed to the function quantileDRM and then ultimately to the function cdfDRM, for estimating the population standard deviations and IQRs required for calculating the default bandwidth for the kernel density estimator. See quantileDRM and cdfDRM for details. The default value is TRUE.

`...` further arguments to be passed to the `density` function, which performs kernel density estimation in R. See `help(density)` for details. One should not pass "x" and "weights" arguments to the `density` function, since those are supposed to be extracted automatically from the fitted DRM object, drmfit.

Details

Note that the default bandwidth for this density estimator is set as that described in Chen and Liu (2013):

$$1.06 n_{total}^{-1/5} \min(\sigma_k, \mathrm{IQR}_k/1.34)$$

, where n_{total} is the total sample size, and $sigma_k$ and IQR_k are the standard deviation and inter-quartile range of the estimated CDF F_k, respectively.

If bandwidth (bw) is not specfied as a ... argument passing to the R functin `density`, the default bandwidth (as described above) will be calculated. That calculation involves the estimation of population quartiles. In this situation, the arguments "interpolation" is passed to `quantileDRM` for quartile estimations. See `quantileDRM` for details.

Value

An output from the `density` function, usually an object with class ’"density"’. See `help(density)` for details.

References

K. Fokianos (2004), Merging information for semiparametric density estimation. *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, 66(4):941-958.

J. Chen and Y. Liu (2013), Quantile and quantile-function estimations under density ratio model. *The Annals of Statistics*, 41(3):1669-1692.

Examples

```
# Data generation
set.seed(25)
n_samples <- c(100, 200, 180, 150, 175)  # sample sizes
x0 <- rgamma(n_samples[1], shape=5, rate=1.8)
x1 <- rgamma(n_samples[2], shape=12, rate=1.2)
x2 <- rgamma(n_samples[3], shape=12, rate=1.2)
x3 <- rgamma(n_samples[4], shape=18, rate=5)
x4 <- rgamma(n_samples[5], shape=25, rate=2.6)
x <- c(x0, x1, x2, x3, x4)

# Fit a DRM with the basis function q(x) = (x, log(abs(x))),
# which is the basis function for gamma family. This basis
# function is the built-in basis function 6.
drmfit <- drmdel(x=x, n_samples=n_samples, basis_func=6)

# Estimate the density of population 3 under the DRM
dens_pop3 <- densityDRM(k=3, drmfit=drmfit)

# Plot the estimated density
```

```
plot(dens_pop3, main=bquote(F[3]), ylim=range(c(0, 0.5)))

# Add the empirical kernel density estimation curve of F_3
# based on the third sample on the above density plot
lines(density(x3), col="blue", lty="28F8")

# Add the true density curve of F_3 on the above density
# plot
lines(seq(min(dens_pop3$y), max(dens_pop3$x), 0.01),
      dgamma(seq(min(dens_pop3$y), max(dens_pop3$x), 0.01),
             18, 5),
      type="l", col="red", lty="dotted")

legend(9, 0.5,
       legend=c("DRM density estimator",
                "Empirical kernel density estimator",
                "True density"),
       col=c("black", "blue", "red"),
       lty=c("solid", "28F8", "dotted"))
```

drmdel *Fit a density ratio model*

Description

Fit a semiparametric density ratio model (DRM) to m+1 (m>=1) samples using maximum dual empirical likelihood method.

Denote the population cumulative distribution functions of the m+1 samples as $F_k(x)$'s, $k = 0, 1, \ldots, m$. We pick $F_0(x)$ as a baseline distribution. The DRM assumes that the ratio of the density of each non-baseline distribution to the density of the baseline distribution satisfies

$$dF_k(x)/dF_0(x) = \exp(\alpha + \beta^T q(x)),\ k = 1, \ldots, m$$

where $q(x)$ is a pre-specified d-dimensional basis function of data, and α, β are model parameters. No parametric form for baseline distribution F_0 is assumed.

Usage

```
drmdel(x, n_samples, basis_func, g_null=NULL,
       g_null_jac=NULL, par_dim_null=NULL, ...)
```

Arguments

`x` a vector formed by concatenating multiple samples, x_0, x_1, ..., x_m, in the order of baseline sample (x_0), non-baseline sample 1 (x_1), ..., non-baseline sample m (x_m).

`n_samples` a vector of length m+1 specifying the sizes of the multiple samples, in the order of 0, 1, ..., m.

`basis_func` basis function q(x) of the DRM; must either be an integer between 1 and 11 or a function of the data, x. The integers represents built-in basis-functions:

1: $q(x) = x$.
2: $q(x) = \log(|x|)$.
3: $q(x) = \sqrt{|x|}$.
4: $q(x) = x^2$.
5: $q(x) = (x,\, x^2)$; Normal model.
6: $q(x) = (x,\, \log(|x|))$; Gamma model.
7: $q(x) = (\log(|x|),\, \sqrt{|x|},\, x)$.
8: $q(x) = (\log(|x|),\, \sqrt{|x|},\, x^2)$.
9: $q(x) = (\log(|x|),\, x,\, x^2)$.
10: $q(x) = (\sqrt{|x|},\, x,\, x^2)$.
11: $q(x) = (\log(|x|),\, \sqrt{|x|},\, x,\, x^2)$.

If the basis function one wants to use is in the above list, one should use the built-in function to maximize the speed of model fitting.

`g_null` the function specifying the null hypothesis about DRM parameter β if there is one; The default is NULL.

`g_null_jac` a funciton specifying the Jacobian matrix of g_null, which must return a matrix of dimension m*d by dim(par_null), if available. The default is NULL.

`par_dim_null` dimension of the parameter vector in null hypothesis if there is one. The default is NULL. If one is carrying out a hypothesis testing problem with a fully specified null hypothesis, one should specify g_null_jac=NULL and par_dim_null=0.

`...` further arguments to be passed to the R function `optim` for maximizing the dual empirical likelihood. See `help(optim)` for details. In the `drmdel` function, by default, the "control$method" and "control$maxit" arguments of `optim` are set to "BFGS" and 10000, respectively.

Value

`drm_info` a list of basic information about the DRM:

m: number of samples - 1.

d: dimension of the basis function.

n_smaples: the input vector of length m+1 specifying the size of each sample.

n_total: total sample size.

basis_func: the input basis function of the DRM.

rho: sample proportion: n_samples/n_total.

`mele` maximum empirical likelihood estimator (MELE) of the model parameters. The output is a vector organized in the following form:

$$(\alpha_1,\, \beta_{1,1},\, \beta_{1,2},\, ...,\, \beta_{1,d},\, \alpha_2,\, \beta_{2,1},\, \beta_{2,2},\, ...,\, \beta_{2,d},\, ...,\, \alpha_m,\, \beta_{m,1},\, \beta_{m,2},\, ...,\, \beta_{m,d}).$$

`info_mat` estimated information matrix.

`negldl` negative log dual empirical likelihood evaluated at mele.

`mele_null` mele of the parameters under the null hypothesis, if available.

`negldl_null` negative log dual empirical likelihood evaluated at mele under the null hypothesis, if available.

`delr` the value of the dual empirical likelihood ratio statistic. If no null hypotheis (g_null) is given, this is simply -2*negldl.

`df` degrees of freedom of the chi-square limiting distribution for DELR statistic under the null.

`p_val` p-value of the DELR test.

`p_est` estimated $dF_k(x)$'s at the observed data points, under the DRM. This is a data frame with the following three columns:

k: label for the populations, k = 0, 1, ..., m.

x: data points; at which 'x' value $dF_k(x)$ is estimated.

p_est: estimated $dF_k(x)$.

NOTE: To estimate the density of $F_k(x)$, it is recommended to use `densityDRM` function.

`cdf_est` estimated CDFs, $F_k(x)$'s, at the observed data points, under the DRM. This is a data frame with the following three columns:

k: label for the populations, k = 0, 1, ..., m.

x: data points; at which 'x' value $F_k(x)$ is estimated.

cdf_est: estimated $F_k(x)$.

NOTE: To estimate CDF $dF_k(x)$, it is recommended to use `cdfDRM` function instead of looking at this output.

References

S. Cai, J. Chen and J. V. Zidek (2014), Hypothesis testing in the presence of multiple samples under density ratio models. Eprint, *arXiv*:1309.4740

A. Keziou and S. Leoni-Aubin (2008), On empirical likelihood for semiparametric two-sample density ratio models. *Journal of Statistical Planning and Inference*, 138:915-928.

Examples

```
# Data generation
set.seed(25)
n_samples <- c(100, 200, 180, 150, 175)  # sample sizes
x0 <- rgamma(n_samples[1], shape=5, rate=1.8)
x1 <- rgamma(n_samples[2], shape=12, rate=1.2)
x2 <- rgamma(n_samples[3], shape=12, rate=1.2)
x3 <- rgamma(n_samples[4], shape=18, rate=5)
x4 <- rgamma(n_samples[5], shape=25, rate=2.6)
x <- c(x0, x1, x2, x3, x4)

# Fit a DRM with the basis function q(x) = (x, log(abs(x))),
# which is the basis function for gamma family.

# There are 11 built-in basis function in drmdel(). And q(x)
# = (x, log(abs(x))) is the 6th basis function, so we can
# fit the model by specifying basis_func=6 in drmdel() as
# follows:
drmfit <- drmdel(x=x, n_samples=n_samples, basis_func=6)
names(drmfit)

# A brief summary of the DRM fit
summaryDRM(drmfit)

# Another way of specifying basis function for drmdel() is
# to pass a user-specified R function to the basis_func
# argument of the drmdel() function.
# NOTE: If the basis function one wants to use is included
# in the built-in function list, one should use the built-in
# functions by passing an integer between 1 to 11 to the
# drmdel() function, because the computation will be faster
# with a built-in function than with a user-specified
# function.
basis_gamma <- function(x) return(c(x, log(abs(x))))
drmfit1 <- drmdel(x=x, n_samples=n_samples,
                  basis_func=basis_gamma)

# One can see the summary of this DRM fit is exactly the
# same as that of the previous fit with basis_func=6
summaryDRM(drmfit1)
```

meleCov *Estimate the covariance matrix of the MELE*

Description

Estimate the covariance matrix of the maximum empirical likelihood estimator (MELE).

Usage

```
meleCov(drmfit)
```

Arguments

drmfit	a fitted DRM object (an output from the drmdel function). See drmdel for details.

Value

The estimated covariance matrix of the MELE.

Examples

```
# Data generation
set.seed(25)
n_samples <- c(100, 200, 180, 150, 175)  # sample sizes
x0 <- rgamma(n_samples[1], shape=5, rate=1.8)
```

```
x1 <- rgamma(n_samples[2], shape=12, rate=1.2)
x2 <- rgamma(n_samples[3], shape=12, rate=1.2)
x3 <- rgamma(n_samples[4], shape=18, rate=5)
x4 <- rgamma(n_samples[5], shape=25, rate=2.6)
x <- c(x0, x1, x2, x3, x4)

# Fit a DRM with the basis function q(x) = (x, log(abs(x))),
# which is the basis function for gamma family. This basis
# function is the built-in basis function 6.
drmfit <- drmdel(x=x, n_samples=n_samples, basis_func=6)

# Check MELE
drmfit$mele

# Estimate the covariance matrix of the MELE
meleCov(drmfit)
```

quantileCompWald — *Wald test for pairwise comparison and linear hypotheses about quantiles under the DRM*

Description

Suppose we have m+1 samples, labeled as $0,\ 1,\ \ldots,\ m$, whose population distributions satisfy the density ratio model (DRM) (see drmdel for the definition of DRM). We now want to test the linear hypothesis about a vector of quantiles $q = (q_1, q_2, \ldots, q_s)^T$ of probably different populations:

$$H_0:\ Aq = b \ \text{ against } \ H_1:\ Aq \neq b,$$

where A is a $t \times s$, $t \leq s$, non-singular matrix and b is a vector. The quantileCompWald function performs a Wald-test for the above hypothesis and also pairwise comparisons of the population quantiles.

Usage

```
quantileCompWald(quantileDRMObject, n_total, pairwise=TRUE,
                 p_adj_method="none", A=NULL, b=NULL)
```

Arguments

`quantileDRMObject`
: an output from the quantileDRM function. It must be a list containing a vector of quantile estimates (quantileDRMobject$est) and a estimated covariance matrix of the quantile estimates (quantileDRMobject$cov). That is, the argument ’cov’ must be set to ’TRUE’ when running quantileDRM.

`n_total`
: total sample size.

`pairwise`
: a logical variable specifying whether to perform pairwise comparisons of the quantiles. The default is TRUE.

`p_adj_method` when pairwise=TRUE, how should the p-values be adjusted for multiple comparisons. The available methods are: "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr" and "none". See `help(p.adjust)` for details. The default is "none", i.e., no adjustment.

`A` the left-hand side t by s, t $\leq$ s, matrix in the linear hypothesis.

`b` the right-hand side t-dimensional vector in the linear hypothesis.

Details

Denote the EL quantile estimate of the q vector as $\hat{q}$, and the estimate of the corresponding covariance matrix as $\hat{\Sigma}$. $\hat{q}$ and $\hat{\Sigma}$ can be calculated using function `quantileDRM` with 'cov=TRUE'.

It is known that, $\sqrt{n_{total}}(\hat{q} - q)$ converges in distribution to a normal distribution with 0 mean and covariance matrix Σ. Also, $\hat{\Sigma}$ is a consistent estimator of Σ. Hence, under the null of the linear hypothesis,

$$n_t otal(A\hat{q} - b)^T \left(A\hat{\Sigma}A^T\right)^{-1}(A\hat{q} - b)$$

has a chi-square limiting distribution with t (=ncol(A)) degrees of freedom.

Value

`p_val_pair` p-values of pairwise comparisons, in the form of a lower triangular matrix. Available only if argument pairwise=TRUE

`p_val` p-value of the linear hypothesis. Available only if argumen 'A' and 'b' are not NULL.

References

J. Chen and Y. Liu (2013), Quantile and quantile-function estimations under density ratio model. *The Annals of Statistics*, 41(3):1669-1692.

Examples

```
# Data generation
set.seed(25)
n_samples <- c(100, 200, 180, 150, 175)  # sample sizes
x0 <- rgamma(n_samples[1], shape=5, rate=1.8)
x1 <- rgamma(n_samples[2], shape=12, rate=1.2)
x2 <- rgamma(n_samples[3], shape=12, rate=1.2)
x3 <- rgamma(n_samples[4], shape=18, rate=5)
x4 <- rgamma(n_samples[5], shape=25, rate=2.6)
x <- c(x0, x1, x2, x3, x4)

# Fit a DRM with the basis function q(x) = (x, log(abs(x))),
# which is the basis function for gamma family. This basis
# function is the built-in basis function 6.
drmfit <- drmdel(x=x, n_samples=n_samples, basis_func=6)

# Quantile comparisons
# Compare the 5^th percentile of population 0, 1, 2 and 3.
# Estimate these quantiles first
qe <- quantileDRM(k=c(0, 1, 2, 3), p=0.05, drmfit=drmfit)

# Create a matrix A and a vector b for testing the equality
# of all these 5^th percentiles. Note that, for this test,
# the contrast matrix A is not unique.
A <- matrix(rep(0, 12), 3, 4)
A[1,] <- c(1, -1, 0, 0)
A[2,] <- c(0, 1, -1, 0)
A[3,] <- c(0, 0, 1, -1)
b <- rep(0, 3)

# Quantile comparisons
# No p-value adjustment for pairwise comparisons
(qComp <- quantileCompWald(qe, n_total=sum(n_samples), A=A,
                           b=b))

# Adjust the p-values for pairwise comparisons using the
# "holm" method.
(qComp1 <- quantileCompWald(qe, n_total=sum(n_samples),
                            p_adj_method="holm", A=A, b=b))
```

quantileDRM *Estimate the quantiles of the populations under the DRM*

Description

Suppose we have m+1 samples, labeled as $0, 1, \ldots, m$, whose population distributions satisfy the density ratio model (DRM) (see drmdel for the definition of DRM). The quantileDRM function estimates the quantiles of the population distributions.

Usage

```
quantileDRM(k, p, drmfit, cov=TRUE, interpolation=TRUE,
             adj=FALSE, adj_val=NULL, bw=NULL, show_bw=FALSE)
```

Arguments

k a vector of labels of populations whose quantiles are to be estimated, with k[i] = 0, 1, ..., m. It could also be a single integer value (in the set of 0, 1, ..., m), in which case, it means that we estimate the quantile of the same population at each of the probability value p (see below for the explaination of argument "p").

p a vector of probabilities (the same length as argument "k") at which the quantiles are estimated; It could also be a single value, in which case, it means that for each population k, we estimate the quantile at a same probability value.

drmfit a fitted DRM object (an output from the drmdel function). See drmdel for details.

`cov` a logical variable specifying whether to estimate the covariance matrix of the quantile estimators. The default is TRUE.

`interpolation` The EL quantile estimator is based on the EL CDF estimator. Hence the way the EL CDF estimate is calculated affects the result of the quantile estimation. This argument is to be passed to the `cdfDRM` function for tweaking the EL CDF estimator. See `cdfDRM` for details.

`adj` a logical variable specifying whether to adjust the CDF estimation by adding a term when estimating quantiles; The default is FALSE. See "Details" section.

`adj_val` a vector of the same length as the argument "k" (or as the argument "p" if the length of "k" is 1) containing the values of adjustment terms for lower or higher quantile estimation, if `adj=TRUE`. The default value, NULL, uses $-1/(2n_k[i])$, where $n_k[i]$ is the size of the k^th sample sample, for each i, to adjust the EL quantile estimator for lower quantile estimation. See "Details" section

`bw` a vector of bandwidths (the same length as the argument "k") for kernel density estimation required for estimating the covariance matrix of the quantile estimators; It could also be a single value, in which case, it means that for each population k, we use the same bandwidth. The default bandwidth, NULL, uses that described in Chen and Liu (2013).

`show_bw` a logical variable specifying whether to output bandwidths when argument cov=TRUE. The default is FALSE.

Details

Denote the estimated CDF of the k^th population as $\hat{F}_k(x)$. The p^th quantile of $F_k(x)$ then is estimated as

$$\inf\{x : \hat{F}_k(x) \geq p\}.$$

The estimated CDF $\hat{F}_k(x)$ reaches its maximum value, 1, at the largest observed data point. If the true CDF $F_k(x)$ is continuous, $F_k(x)$ tends to 1 when x tends to infinity. Hence, when estimate an upper quanitle, say 0.95^th quantile, of F_k, the quantile estimator is likely to under estimate the ture quantile, especially when sample size is not too large. To adjust an upper quantile estimator for possible under-estimation, one may want to adjust the estimated CDF as

$$\hat{F}_k(x) + \text{adj\_val}.$$

and use the adjusted CDF to estimate quantiles. To make an upper quantile estimator larger, the adj_val should have a positive value. Similarly, to adjust lower quantile estimates for possible over-estimation, the adj_val should have a negative value.

The `quantileDRM` function, by default, does not adjust CDF estimators (`adjust=FALSE`). When `adjust=TRUE`, the default adj_val is set to $-1/(2n_k[i])$, where $n_k[i]$ is the size of the k^th sample sample, for each i, to adjust the EL quantile estimator for lower quantile estimation.

Value

`est` quantile estimates.

`cov` estimated covariance matrix of the quantile estimators, available only if argument cov=TRUE.

`bw` bandwidths used for kernel density estimation required for estimating the covariance matrix of the quantile estimators, available only if argument cov=TRUE and show_bw=TRUE.

References

J. Chen and Y. Liu (2013), Quantile and quantile-function estimations under density ratio model. *The Annals of Statistics*, 41(3):1669-1692.

Examples

```
# Data generation
set.seed(25)
n_samples <- c(100, 200, 180, 150, 175)  # sample sizes
x0 <- rgamma(n_samples[1], shape=5, rate=1.8)
x1 <- rgamma(n_samples[2], shape=12, rate=1.2)
x2 <- rgamma(n_samples[3], shape=12, rate=1.2)
x3 <- rgamma(n_samples[4], shape=18, rate=5)
x4 <- rgamma(n_samples[5], shape=25, rate=2.6)
x <- c(x0, x1, x2, x3, x4)

# Fit a DRM with the basis function q(x) = (x, log(abs(x))),
# which is the basis function for gamma family. This basis
# function is the built-in basis function 6.
drmfit <- drmdel(x=x, n_samples=n_samples, basis_func=6)

# Quantile estimation
# Denote the p^th quantile of the k^th, k=0, 1, ..., 4,
# population as q_{k,p}.

# Estimate q_{0,0.25}, q_{0,0.6}, q_{1,0.1} and q_{2,0.1}.
(qe <- quantileDRM(k=c(0, 0, 1, 2), p=c(0.25, 0.6, 0.1, 0.1),
                  drmfit=drmfit))

# Estimate the 0.05^th, 0.2^th and 0.8^th quantiles of F_3
(qe1 <- quantileDRM(k=3, p=c(0.05, 0.2, 0.8), drmfit=drmfit))

# Estimate the 0.05^th quantiles of F_1, F_3 and F_4
(qe2 <- quantileDRM(k=c(1 , 3, 4), p=0.05, drmfit=drmfit))
```

`summaryDRM` *Print simple summaries of a DRM fit object*

Description

Print simple summaries of a density ratio model (DRM) fit object, i.e. an output from `drmdel` function.

Usage

```
summaryDRM(drmfit)
```

Arguments

drmfit a fitted DRM object (an output from the drmdel function). See drmdel for details.

Value

Summaries of the fitted DRM object, including the basic information about the fitted DRM, maximum empirical likelihood estimates (MELE) of the DRM parameters, results of the dual empirical likelihood ratio (DELR) test for hypothesis about the DRM parameter β and some summary statistics of the estimated distribution functions F_k, $k = 0, 1, ..., m$.

Note

For more detailed but less user-friendly formatted information about a fitted DRM object, one should directly check the output from the drmdel function. See drmdel for details.

Examples

```
# Data generation
set.seed(25)
n_samples <- c(100, 200, 180, 150, 175)  # sample sizes
x0 <- rgamma(n_samples[1], shape=5, rate=1.8)
x1 <- rgamma(n_samples[2], shape=12, rate=1.2)
x2 <- rgamma(n_samples[3], shape=12, rate=1.2)
x3 <- rgamma(n_samples[4], shape=18, rate=5)
x4 <- rgamma(n_samples[5], shape=25, rate=2.6)
x <- c(x0, x1, x2, x3, x4)

# Fit a DRM with the basis function q(x) = (x, log(abs(x))),
# which is the basis function for gamma family. This basis
# function is the built-in basis function 6.
drmfit <- drmdel(x=x, n_samples=n_samples, basis_func=6)

# A brief summary of the DRM fit
summaryDRM(drmfit)

# List the details of the DRM fit.  See help(drmdel) for
# details.
names(drmfit)
```

Index