Package ‘emlik2’

October 13, 2022

Version 1.32

Title Empirical Likelihood Ratio Test for Two Samples with Censored Data

Author William H. Barton <williamhbarton1@gmail.com> under the supervision of Dr. Mai Zhou

Maintainer Mai Zhou <maizhou@gmail.com>

Depends R (>= 3.2.5)

Imports stats

Description
Calculates the p-value for a mean-type hypothesis (or multiple mean-type hypotheses) based on two samples with possible censored data.

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2022-01-04 10:50:02 UTC

R topics documented:

- el2.cen.EMm ............................................................ 1
- el2.cen.EMs ............................................................. 4
- el2.test.wtm ............................................................ 7
- el2.test.wts ............................................................ 10

Index

el2.cen.EMm Computes p-value for multiple mean-type hypotheses, based on two independent samples that may contain censored data.
Description

This function uses the EM algorithm to calculate a maximized empirical likelihood ratio for a set of \( p \) hypotheses as follows:

\[
H_0 : E(g(x, y) - mean) = 0
\]

where \( E \) indicates expected value; \( g(x, y) \) is a vector of user-defined functions \( g_1(x, y), \ldots, g_p(x, y) \); and \( mean \) is a vector of \( p \) hypothesized values of \( E(g(x, y)) \). The two samples \( x \) and \( y \) are assumed independent. They may be uncensored, right-censored, left-censored, or left-and-right (“doubly”) censored. A p-value for \( H_0 \) is also calculated, based on the assumption that \(-2\log(\text{empirical likelihood ratio})\) is asymptotically distributed as chisq\((p)\).

Usage

\[
el2.cen.EMm(x, dx, wx=rep(1,length(x)), y, dy, wy=rep(1,length(y)), p, H, xc=1:length(x), yc=1:length(y), mean, maxit=15)
\]

Arguments

- \( x \): a vector of the data for the first sample
- \( dx \): a vector of the censoring indicators for \( x \): 0=right-censored, 1=uncensored, 2=left-censored
- \( wx \): a vector of data case weight for \( x \)
- \( y \): a vector of the data for the second sample
- \( dy \): a vector of the censoring indicators for \( y \): 0=right-censored, 1=uncensored, 2=left-censored
- \( wy \): a vector of data case weight for \( y \)
- \( p \): the number of hypotheses
- \( H \): a matrix defined as \( H = [H_1, H_2, \ldots, H_p] \), where \( H_k = [g_k(x_i, y_j) - mu_k], k = 1, \ldots, p \)
- \( xc \): a vector containing the indices of the \( x \) datapoints, controls if tied \( x \) collapse or not
- \( yc \): a vector containing the indices of the \( y \) datapoints, ditto
- \( mean \): the hypothesized value of \( E(g(x, y)) \)
- \( maxit \): a positive integer used to control the maximum number of iterations of the EM algorithm; default is 15

Details

The value of \( mean_k \) should be chosen between the maximum and minimum values of \( g_k(x_i, y_j) \); otherwise there may be no distributions for \( x \) and \( y \) that will satisfy \( H_0 \). If \( mean_k \) is inside this interval, but the convergence is still not satisfactory, then the value of \( mean_k \) should be moved closer to the NPMLE for \( E(g_k(x, y)) \). (The NPMLE itself should always be a feasible value for \( mean_k \).)
Value

e12.cen.EMm returns a list of values as follows:

- **xd1**: a vector of unique, uncensored x-values in ascending order
- **yd1**: a vector of unique, uncensored y-values in ascending order
- **temp3**: a list of values returned by the e12.test.wtm function (which is called by e12.cen.EMm)
- **mean**: the hypothesized value of $E(g(x,y))$
- **NPMLE**: a non-parametric-maximum-likelihood-estimator vector of $E(g(x,y))$
- **logel00**: the log of the unconstrained empirical likelihood
- **logel**: the log of the constrained empirical likelihood
- **"-2LLR"**: $-2*(\text{log-likelihood-ratio})$ for the p simultaneous hypotheses
- **Pval**: the p-value for the p simultaneous hypotheses, equal to $1 - \text{pchisq}(-2\text{LLR}, \text{df} = p)$
- **logvec**: the vector of successive values of logel computed by the EM algorithm (should converge toward a fixed value)
- **sum.muvec**: sum of the probability jumps for the uncensored x-values, should be 1
- **sum.nuvec**: sum of the probability jumps for the uncensored y-values, should be 1

Author(s)

William H. Barton <bbarton@lexmark.com>

References


Examples

```r
x<-c(10, 80, 273, 279, 324, 391, 566, 85, 852, 881, 895, 954, 1101, 1133, 1393, 1444, 1513, 1585, 1669, 1823, 1941)
dx<-c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0)
y<-c(21, 38, 39, 51, 77, 185, 240, 289, 524, 610, 677, 798, 881, 899, 946, 1010, 1074, 1147, 1154, 1269, 1329, 1484, 1493, 1559, 1602, 1684, 1900, 1952)
dy<-c(1,1,1,1,2,2,1,1,1,1,2,1,1,1,1,0,0,1,1,0,0,1,0,0,0)
nx<-length(x)
ny<-length(y)
xc<-1:nx
yc<-1:ny
wx<-rep(1,nx)
wy<-rep(1,ny)
mu=c(0.5,0.5)
p <- 2
H1<-matrix(NA,nrow=nx,ncol=ny)
H2<-matrix(NA,nrow=nx,ncol=ny)
for (i in 1:nx) {
  for (j in 1:ny) {
    H1[i,j]<-(x[i]>y[j])
    H2[i,j]<-(x[i]>1060)
  }
}
H=matrix(c(H1,H2),nrow=nx,ncol=p*ny)
# Ho1: X is stochastically equal to Y
# Ho2: mean of X equals mean of Y
el2.cen.EMm(x=x, dx=dx, y=y, dy=dy, p=2, H=H, mean=mu, maxit=10)
# Result: Pval is 0.6310234, so we cannot with 95 percent confidence reject the two
# simultaneous hypotheses Ho1 and Ho2
```

el2.cen.EMs

Computes p-value for a single mean-type hypothesis, based on two independent samples that may contain censored data.

Description

This function uses the EM algorithm to calculate a maximized empirical likelihood ratio for the hypothesis

\( H_0 : E(g(x, y) - \text{mean}) = 0 \)

where \( E \) indicates expected value; \( g(x, y) \) is a user-defined function of \( x \) and \( y \); and \( \text{mean} \) is the hypothesized value of \( E(g(x, y)) \). The samples \( x \) and \( y \) are assumed independent. They may be uncensored, right-censored, left-censored, or left-and-right (“doubly”) censored. A p-value for \( H_0 \) is also calculated, based on the assumption that -2*\log(empirical likelihood ratio) is approximately distributed as chisq(1).
Usage

\texttt{el2.cen.EMs(x, dx, y, dy, fun=function(x, y)\{x>=y\}, mean=0.5, maxit=25)}

Arguments

\texttt{x} \quad \text{a vector of the data for the first sample}

\texttt{dx} \quad \text{a vector of the censoring indicators for } x: 0=\text{right-censored}, 1=\text{uncensored}, 2=\text{left-censored}

\texttt{y} \quad \text{a vector of the data for the second sample}

\texttt{dy} \quad \text{a vector of the censoring indicators for } y: 0=\text{right-censored}, 1=\text{uncensored}, 2=\text{left-censored}

\texttt{fun} \quad \text{a user-defined, continuous-weight-function } g(x, y) \text{ used to define the mean in the hypothesis } H_0. \text{ The default is } \texttt{fun=function(x, y)\{x>=y\}}.

\texttt{mean} \quad \text{the hypothesized value of } E(g(x, y)); \text{ default is 0.5}

\texttt{maxit} \quad \text{a positive integer used to set the number of iterations of the EM algorithm; default is 25}

Details

The value of \texttt{mean} should be chosen between the maximum and minimum values of \(g(x_i, y_j)\); otherwise there may be no distributions for \(x\) and \(y\) that will satisfy \(H_0\). If \texttt{mean} is inside this interval, but the convergence is still not satisfactory, then the value of \texttt{mean} should be moved closer to the NPMLE for \(E(g(x, y))\). (The NPMLE itself should always be a feasible value for \texttt{mean}.)

Value

\texttt{el2.cen.EMs} returns a list of values as follows:

\texttt{xd1} \quad \text{a vector of the unique, uncensored } x\text{-values in ascending order}

\texttt{yd1} \quad \text{a vector of the unique, uncensored } y\text{-values in ascending order}

\texttt{temp3} \quad \text{a list of values returned by the \texttt{el2.test.wts} function (which is called by \texttt{el2.cen.EMs})}

\texttt{mean} \quad \text{the hypothesized value of } E(g(x, y))

\texttt{funNPMLE} \quad \text{the non-parametric-maximum-likelihood-estimator of } E(g(x, y))

\texttt{logel00} \quad \text{the log of the unconstrained empirical likelihood}

\texttt{logel} \quad \text{the log of the constrained empirical likelihood}

\texttt{"-2LLR"} \quad -2*(\texttt{logel-\texttt{logel00}})

\texttt{Pval} \quad \text{the estimated } p\text{-value for } H_0, \text{ computed as } 1 - \texttt{pchisq(-2LLR, df = 1)}

\texttt{logvec} \quad \text{the vector of successive values of } \texttt{logel} \text{ computed by the EM algorithm (should converge toward a fixed value)}

\texttt{sum_muvec} \quad \text{sum of the probability jumps for the uncensored } x\text{-values, should be 1}

\texttt{sum_nuvec} \quad \text{sum of the probability jumps for the uncensored } y\text{-values, should be 1}

\texttt{constraint} \quad \text{the realized value of } \sum_{i=1}^n \sum_{j=1}^m (g(x_i, y_j) - \texttt{mean}) \mu_i \nu_j, \text{ where } \mu_i \text{ and } \nu_j \text{ are the probability jumps at } x_i \text{ and } y_j, \text{ respectively, that maximize the empirical likelihood ratio. The value of } \texttt{constraint} \text{ should be close to 0.}
Author(s)

William H. Barton <bbarton@lexmark.com>

References


Examples

```r
x<-c(10,80,209,273,279,324,391,415,566,785,852,881,895,954,1101,1133,1337,1393,1408,1444,1513,1585,1669,1823,1941)
dx<-c(1,2,1,1,1,2,1,1,1,1,1,1,0,1,0,0,0,1,1,0)
dy<-c(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,0,0,0,0,0,0,0)

# Ho1: X is stochastically equal to Y
el2.cen.EMs(x, dx, y, dy, fun=function(x,y){x>=y}, mean=0.5, maxit=25)
# Result: Pval = 0.7090658, so we cannot with 95 percent confidence reject Ho1

# Ho2: mean of X equals mean of Y
el2.cen.EMs(x, dx, y, dy, fun=function(x,y){x-y}, mean=0.5, maxit=25)
# Result: Pval = 0.9695593, so we cannot with 95 percent confidence reject Ho2
```
el2.test.wtm

Computes maximum-likelihood probability jumps for multiple mean-type hypotheses, based on two independent uncensored samples

Description

This function computes the maximum-likelihood probability jumps for multiple mean-type hypotheses, based on two samples that are independent, uncensored, and weighted. The target function for the maximization is the constrained log(empirical likelihood) which can be expressed as,

$$
\sum_{dx_i=1} wx_i \log \mu_i + \sum_{dy_j=1} wy_j \log \nu_j - \eta(1 - \sum_{dx_i=1} \mu_i) - \delta(1 - \sum_{dy_j=1} \nu_j) - \lambda(\mu^T H_1 \nu_1, \ldots, \mu^T H_p \nu)^T
$$

where the variables are defined as follows:

- $x$ is a vector of uncensored data for the first sample
- $y$ is a vector of uncensored data for the second sample
- $wx$ is a vector of estimated weights for the first sample
- $wy$ is a vector of estimated weights for the second sample
- $\mu$ is a vector of estimated probability jumps for the first sample
- $\nu$ is a vector of estimated probability jumps for the second sample
- $H_k = [g_k(x_i, y_j) - mean_k], k = 1, \ldots, p$, where $g_k(x, y)$ is a user-chosen function
- $H = [H_1, \ldots, H_p]$ (used as argument in el.cen.EMm function, which calls el2.test.wtm)
- $mean$ is a vector of length $p$ of hypothesized means, such that $mean_k$ is the hypothesized value of $E(g_k(x, y))$
- $E$ indicates “expected value”

Usage

```
el2.test.wtm(xd1, yd1, wxd1new, wyd1new, muvec, nuvec, Hu, Hmu, Hnu, p, mean, maxit=15)
```

Arguments

- **xd1**: a vector of uncensored data for the first sample
- **yd1**: a vector of uncensored data for the second sample
- **wxd1new**: a vector of estimated weights for xd1
- **wyd1new**: a vector of estimated weights for yd1
- **muvec**: a vector of estimated probability jumps for xd1
- **nuvec**: a vector of estimated probability jumps for yd1
- **Hu**: $H_u = [H_1 - [mean_1], \ldots, H_p - [mean_p]], dx_i = 1, dy_j = 1$
- **Hmu**: a matrix, whose calculation is shown in the example below
- **Hnu**: a matrix, whose calculation is shown in the example below
p the number of hypotheses
mean a vector of hypothesized values of $E(g_k(u,v)), k = 1, \ldots, p$
maxit a positive integer used to control the maximum number of iterations in the Newton-Raphson algorithm; default is 10

Details
This function is called by el2.cen.EMm. It is listed here because the user may find it useful elsewhere.

The value of mean$_k$ should be chosen between the maximum and minimum values of $g_k(xd_1, yd_1)$; otherwise there may be no distributions for $xd_1$ and $yd_1$ that will satisfy the the mean-type hypothesis. If mean$_k$ is inside this interval, but the convergence is still not satisfactory, then the value of mean$_k$ should be moved closer to the NPMLE for $E(g(xd_1, yd_1))$. (The NPMLE itself should always be a feasible value for mean$_k$.) The calculations for this function are derived in Owen (2001).

Value
el2.test.wtm returns a list of values as follows:

constmat a matrix of row vectors, where the $k$th row vector holds successive values of $\mu^T H_k \nu, k = 1, \ldots, p$, generated by the Newton-Raphson algorithm
lam the vector of Lagrangian multipliers used in the calculations
muvec1 the vector of probability jumps for the first sample that maximize the weighted empirical likelihood
nuvec1 the vector of probability jumps for the second sample that maximize the weighted empirical likelihood
mean the vector of hypothesized means

Author(s)
William H. Barton <bbarton@lexmark.com>

References

Examples

#Ho1: P(X>Y) = 0.5
#Ho2: P(X>1060) = 0.5
#g1(x) = I[x > y]
#g2(y) = I[x > 1060]
mean<-c(0.5,0.5)
p<-2
xd1<-c(10,85,209,273,279,324,391,566,881,895,954,1101,1393,1669,1823,1941)
nx1=length(xd1)
yd1<-c(21,38,39,51,77,185,524,610,612,677,798,899,1010,1074,1147,1154,1329,1484,1602,1952)
ny1=length(yd1)

wxd1new<-c(2.267983, 1.123600, 1.121683, 1.121683, 1.121683, 1.121683, 1.121683, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.261740, 2.912753, 2.912753)

muvec<-c(0.08835785, 0.04075290, 0.04012084, 0.04012084, 0.04012084, 0.04012084, 0.04012084, 0.03538020, 0.03389263, 0.03389263, 0.03322693, 0.04901516, 0.05902008, 0.13065491, 0.13065491, 0.13065491)

wyd1new<-c(1.431653, 1.431653, 1.431653, 1.431653, 1.431653, 1.438453, 1.079955, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1.222883, 1.227865, 1.739636, 5.809616)

nuvec<-c(0.04249966, 0.04249966, 0.04249966, 0.04249966, 0.04249966, 0.04249966, 0.04316922, 0.03452722, 0.0346312, 0.0346312, 0.0346312, 0.0346312, 0.03300598, 0.03300598, 0.0333333, 0.0333333, 0.03333333, 0.03382827, 0.03382827, 0.04136800, 0.04429270, 0.05992020, 0.22762676)

Hu<-matrix(NA,nrow=nx1,ncol=ny1)
H2u<-matrix(NA,nrow=nx1,ncol=ny1)
for (i in 1:nx1) {
  for (j in 1:ny1) {
    Hu[i,j]<-(xd1[i]>yd1[j])
    H2u[i,j]<-(xd1[i]>1060) }
}

for (k in 1:p) {
  M1 <- matrix(mean[k], nrow=nx1, ncol=ny1)
  Hu[,((k-1)*ny1+1):(k*ny1)] <- Hu[,((k-1)*ny1+1):(k*ny1)] - M1
  Hmu <- matrix(Hu, nrow=p, ncol=ny1*nx1)
  Hnu <- matrix(Hu, nrow=p, ncol=ny1*nx1)
  for (i in 1:p) {
    for (k in 1:nx1) {
      Hmu[i, ((k-1)*ny1+1):(k*ny1)] <- Hu[K,((i-1)*ny1+1):(i*ny1)]
    }
    for (i in 1:p) {
      for (k in 1:ny1) {
        Hnu[i, ((k-1)*nx1+1):(k*nx1)] <- Hu[(1:nx1),(i-1)*ny1+k]
      }
    }
  }
}

el2.test.wtm(xd1,yd1,wxd1new, wyd1new, muvec, nuvec, Hu, Hmu, Hnu, p, mean, maxit=10)

#muvec1
# [1] 0.08835789 0.04075290 0.04012083 0.04012083 0.04012083 0.04012083 0.04012083 0.04012083
# [8] 0.03538021 0.03389264 0.03389264 0.03322693 0.04901516 0.05902008 0.13065491 0.13065491
# [15] 0.13065495 0.13065495 0.13065495

#nuvec1
# [1] 0.04249967 0.04249967 0.04249967 0.04249967 0.04249967 0.04316920 0.03452722
# [8] 0.0346312 0.0346312 0.0346312 0.0346312 0.0346312 0.03300598 0.03300598 0.0333333
# [15] 0.0333333 0.03382827 0.03382827 0.04136801 0.04429270 0.05992020 0.22762677

# $lam
el2.test.wts

Computes maximum-likelihood probability jumps for a single mean-type hypothesis, based on two independent uncensored samples

Description

This function computes the maximum-likelihood probability jumps for a single mean-type hypothesis, based on two samples that are independent, uncensored, and weighted. The target function for the maximization is the constrained log(empirical likelihood) which can be expressed as,

$$ \sum_{x_i=1} wx_i \log \mu_i + \sum_{y_j=1} wy_j \log \nu_j - \eta(1 - \sum_{x_i=1} \mu_i) - \delta(1 - \sum_{y_j=1} \nu_j) - \lambda \sum_{x_i=1} \sum_{y_j=1} (g(x_i, y_j) - \text{mean}) \mu_i \nu_j $$

where the variables are defined as follows:

- $x$ is a vector of data for the first sample
- $y$ is a vector of data for the second sample
- $wx$ is a vector of estimated weights for the first sample
- $wy$ is a vector of estimated weights for the second sample
- $\mu$ is a vector of estimated probability jumps for the first sample
- $\nu$ is a vector of estimated probability jumps for the second sample

Usage

el2.test.wts(u, v, wu, wv, mu0, nu0, indicmat, mean, lamOld=0)

Arguments

- u: a vector of uncensored data for the first sample
- v: a vector of uncensored data for the second sample
- wu: a vector of estimated weights for u
- wv: a vector of estimated weights for v
- mu0: a vector of estimated probability jumps for u
- nu0: a vector of estimated probability jumps for v
- indicmat: a matrix $[g(u_i, v_j) - \text{mean}]$ where $g(u, v)$ is a user-chosen function
- mean: a hypothesized value of $E(g(u, v))$, where $E$ indicates “expected value.”
- lamOld: The previous solution of lambda, used as the starting point to search for new solution of lambda.
Details

This function is called by `el2.cen.EMs`. It is listed here because the user may find it useful elsewhere.

The value of `mean` should be chosen between the maximum and minimum values of \((u_i, v_j)\); otherwise there may be no distributions for \(u\) and \(v\) that will satisfy the the mean-type hypothesis. If `mean` is inside this interval, but the convergence is still not satisfactory, then the value of `mean` should be moved closer to the NPMLE for \(E(g(u, v))\). (The NPMLE itself should always be a feasible value for `mean`). The calculations for this function are derived in Owen (2001).

Value

```
el2.test.wts returns a list of values as follows:

u       the vector of uncensored data for the first sample
wu      the vector of weights for \(u\)
jumpu   the vector of probability jumps for \(u\) that maximize the weighted empirical likelihood
v       the vector of uncensored data for the second sample
wv      the vector of weights for \(v\)
jumpv   the vector of probability jumps for \(v\) that maximize the weighted empirical likelihood
lam     the value of the Lagrangian multiplier found by the calculations
```

Author(s)

William H. Barton <bbarton@lexmark.com> and modified by Mai Zhou.

References


Examples

```
u<-c(10, 209, 273, 279, 324, 391, 566, 785)
v<-c(21, 38, 51, 77, 185, 240, 289, 524)
wu<-c(2.442931, 1.122365, 1.113239, 1.113239, 1.104113, 1.104113, 1.000000, 1.000000)
wv<-c(1, 1, 1, 1, 1, 1, 1, 1)
mu0<-c(0.3774461, 0.1042739, 0.09649724, 0.09649724, 0.08872055, 0.08872055, 0.0739222, 0.0739222)
nu0<-c(0.1013718, 0.1013718, 0.1013718, 0.1013718, 0.1013718, 0.1013718, 0.1013718, 0.1287447, 0.1534831)
mean<-0.5

#let fun=function(x,y){x>=y)
indicmat<-matrix(nrow=8,ncol=9,c(
-0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5,
-0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5,
-0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5,
-0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5,
-0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5,
-0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5,
-0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5,
-0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5,
-0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5,
-0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5,
-0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5,
-0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5,
-0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5,
-0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5,
-0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5,
-0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5,
-0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5,
el2.test.wts

-0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5,
-0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5,
-0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5,
-0.5, -0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5,
-0.5, -0.5, -0.5, 0.5, 0.5, 0.5, 0.5, 0.5,
-0.5, -0.5, -0.5, -0.5, 0.5, 0.5, 0.5, 0.5)

el2.test.wts(u,v,wv,mu0,nu0,indicmat,mean)

# jumpu
# [1] 0.3774461, 0.1042739, 0.09649724, 0.09649724, 0.08872055, 0.08872055, 0.0739222, 0.0739222

# jumpv
# [1] 0.1013718, 0.1013718, 0.1013718, 0.1013718, 0.1013718, 0.1013718, 0.1095413, 0.1287447,
# [9] 0.1534831

# lam
# [1] 7.055471
Index

* nonparametric
  - el2.cen.EMm, 1
  - el2.cen.EMS, 4
  - el2.test.wtm, 7
  - el2.test.wts, 10

el2.cen.EMm, 1
el2.cen.EMS, 4
el2.test.wtm, 7
el2.test.wts, 10