# Package ‘emplik2’

**Version** 1.20  
**Date** 2015-02-04  
**Title** Empirical Likelihood Ratio Test for Two Samples with Censored Data  
**Author** William H. Barton &lt;williamhbarton1@gmail.com&gt; under the supervision of Dr. Mai Zhou &lt;mai@ms.uky.edu&gt;  
**Maintainer** William H. Barton &lt;williamhbarton1@gmail.com&gt;  
**Depends** R (&gt;= 2.15.0)  
**Description** Calculates the p-value for a mean-type hypothesis (or multiple mean-type hypotheses) based on two samples with censored data.  
**License** GPL (&gt;= 2)  
**NeedsCompilation** no  
**Repository** CRAN  
**Date/Publication** 2015-02-05 00:23:32

## R topics documented:

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>el2.cen.EMm</td>
<td>Computes p-value for multiple mean-type hypotheses, based on two independent samples that may contain censored data.</td>
</tr>
<tr>
<td>el2.cen.EMS</td>
<td></td>
</tr>
<tr>
<td>el2.test.wtm</td>
<td></td>
</tr>
<tr>
<td>el2.test.wts</td>
<td></td>
</tr>
</tbody>
</table>

**Index** 13
**Description**

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

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

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 \( \text{mean} \) 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 \(-2\times\log(\text{empirical likelihood ratio})\) is asymptotically distributed as \( \chi^2(p) \).

**Usage**

\[
e12.\text{cen.EMm}(x, dx, y, dy, p, H, xc=1:length(x), yc=1:length(y),
\text{mean}, \text{maxit}=10)
\]

**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
- **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
- **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
- **yc**: a vector containing the indices of the \( y \) datapoints
- **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 10

**Details**

The value of \( \text{mean}_k \) 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 \( \text{mean}_k \) is inside this interval, but the convergence is still not satisfactory, then the value of \( \text{mean}_k \) should be moved closer to the NPMLE for \( E(g_k(x, y)) \). (The NPMLE itself should always be a feasible value for \( \text{mean}_k \).)

**Value**

\( e12.\text{cen.EMm} \) 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*(log-likelihood-ratio) for the p simultaneous hypotheses

Pval the p-value for the p simultaneous hypotheses, equal to \(1 - \text{pchisq}(-2\text{LLR}, \ df = p)\)

logvec the vector of successive values of logel computed by the EM algorithm (should converge toward a fixed value)

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

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

Author(s)

William H. Barton <bbarton@lexmark.com>

References


Zhou, M. (2009) emplik package on CRAN website. Dr. Zhou is my PhD advisor at the University of Kentucky. My e12.cen.EMm function extends Dr. Zhou’s e1.cen.EM2 function from one-sample to two-samples.
Examples

\begin{verbatim}
x<-c(10, 80, 209, 273, 279, 324, 391, 415, 566, 85, 852, 881, 895, 954, 1101, 1133, 1337, 1393, 1408, 1444, 1513, 1585, 1669, 1823, 1941)  
dx<-c(1, 38, 51, 415, 240, 289, 524, 610, 612, 677, 798, 881, 899, 946, 1010, 1074, 1147, 1154, 1199, 1269, 1329, 1484, 1493, 1559, 1602, 1684, 1900, 1952)  
y<-c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)  
dy<-c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)  
nx<-length(x)  
nx<-length(y)
xc<-1:nx  
yc<-1:ny  
wx<-rep(1,nx)  
wy<-rep(1,ny)
maxit<-10
mean=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]>1000)  
  }
  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.EM(x, dx, dy, p, H, xc=1:length(x), yc=1:length(y),
mean, maxit=10)

# Result: Pval is 0.6310234, so we cannot with 95 percent confidence reject the two
# simultaneous hypotheses Ho1 and Ho2
\end{verbatim}

el2.cen.EMS  \hspace{1cm} Computes p-value for a single mean-type hypothesis, based on two
\hspace{1cm} independent samples that may contain censored data.

Description

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

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

where \( E \) indicates expected value; \( g(x,y) \) is a user-defined function of \( x \) and \( y \); and \( mean \) 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 \(-2\times\log(\text{empirical likelihood ratio})\) is approximately distributed as chisq(1).
Usage
el2.cen.EMs(x, dx, y, dy, fun=function(x, y) {x > y}, mean=0.5, maxit=25)

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
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
fun a user-defined, continuous-weight-function \( g(x, y) \) used to define the mean in the hypothesis \( H_0 \). The default is \( \text{fun} = \text{function}(x, y) \{x > y\} \).
mean the hypothesized value of \( E(g(x, y)) \); default is 0.5
maxit a positive integer used to set the number of iterations of the EM algorithm; default is 25

Details

The value of \( \text{mean} \) 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 \( \text{mean} \) is inside this interval, but the convergence is still not satisfactory, then the value of \( \text{mean} \) should be moved closer to the NPMLE for \( E(g(x, y)) \). (The NPMLE itself should always be a feasible value for \( \text{mean} \).)

Value

el2.cen.EMs returns a list of values as follows:

xd1 a vector of the unique, uncensored \( x \)-values in ascending order
yd1 a vector of the unique, uncensored \( y \)-values in ascending order
tempS a list of values returned by the el2.test.wts function (which is called by el2.cen.EMs)
mean the hypothesized value of \( E(g(x, y)) \)
funNPMLE the non-parametric-maximum-likelihood-estimator of \( E(g(x, y)) \)
logel00 the log of the unconstrained empirical likelihood
logel the log of the constrained empirical likelihood
"-2LLR" \(-2 \times \text{logel-logel00}\)
Pval the estimated p-value for \( H_0 \), computed as \( 1 - \text{pchisq}(-2 \text{LLR}, \ df = 1) \)
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
constraint the realized value of \( \sum_{i=1}^n \sum_{j=1}^m (g(x_i, y_j) - \text{mean}) \mu_i \nu_j \), where \( \mu_i \) and \( \nu_j \) are the probability jumps at \( x_i \) and \( y_j \), respectively, that maximize the empirical likelihood ratio. The value of \( \text{constraint} \) should be close to 0.
Author(s)

William H. Barton <bbarton@lexmark.com>

References


Zhou, M. (2009) *emplik* package on CRAN website. Dr. Zhou is my PhD advisor at the University of Kentucky. My el2.cen.EMs function extends Dr. Zhou’s el.cen.EM function from one-sample to two-samples.

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(1,2,1,1,1,1,2,1,1,1,1,1,1,0,0,1,0,0,0,1,1,0)
y<-c(21,38,39,51,77,185,240,289,524,610,612,677,798,881,899,946,
1018,1074,1147,1154,1199,1269,1329,1484,1493,1559,1682,1684,1900,1952)
dy<-c(1,1,1,1,1,1,1,1,1,1,1,1,1,0,1,1,0,0,0,0,0,0)

# 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
- \(H = [H_1, ..., H_p]\) (used as argument in el.cen.Emn 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=10)
```

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
Actual page content:

\( p \) the number of hypotheses

\( \text{mean} \) a vector of hypothesized values of \( E(g_k(u,v)) \), \( k = 1, \ldots, p \)

\( \text{maxit} \) 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 \texttt{el2.cen} and \texttt{emm}. It is listed here because the user may find it useful elsewhere.

The value of \textit{mean}_k should be chosen between the maximum and minimum values of \( g_k(xd1_i, yd1_j) \); otherwise there may be no distributions for \( xd1 \) and \( yd1 \) that will satisfy the the mean-type hypothesis. If \textit{mean}_k is inside this interval, but the convergence is still not satisfactory, then the value of \textit{mean}_k should be moved closer to the NPMLE for \( E(g(xd1, yd1)) \). (The NPMLE itself should always be a feasible value for \textit{mean}_k.) The calculations for this function are derived in Owen (2001).

**Value**

\texttt{el2.test.wtm} returns a list of values as follows:

- \texttt{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
- \texttt{lam} the vector of Lagrangian multipliers used in the calculations
- \texttt{muvec1} the vector of probability jumps for the first sample that maximize the weighted empirical likelihood
- \texttt{nuvec1} the vector of probability jumps for the second sample that maximize the weighted empirical likelihood
- \texttt{mean} the vector of hypothesized means

**Author(s)**

William H. Barton <bbarton@lexmark.com>

**References**


**Examples**

```r
#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
```
H1u <- matrix(NA, nrow=nx1, ncol=ny1)
H2u <- matrix(NA, nrow=nx1, ncol=ny1)
for (i in 1:nx1) {
  for (j in 1:ny1) {
    H1u[i,j] <- c(xd1[i], yd1[j])
    H2u[i,j] <- c(xd1[i]+1000)
  }
}
Hu <- matrix(c(H1u, H2u), nrow=nx1, ncol=p*ny1)
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(NA, nrow=p, ncol=ny1*nx1)
  Hnu <- matrix(NA, nrow=p, ncol=ny1*nx1)
  for (i in 1:p) {
    for (k in 1:nx1) {
      Hmu[i, ((k-1)*ny1+1):((k*ny1)] <- hu[i, ((i-1)*ny1+1):((i*ny1)]
    }
  }
  for (i in 1:nx1) {
    for (k in 1:ny1) {
      Hmu[i, ((k-1)*nx1+1):((k*nx1)] <- hu[i, ((i-1)*nx1+1):((i*nx1+1))
    }
  }
}

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
# [8] 0.03538021 0.03389264 0.03389264 0.03389264 0.03322693 0.04901513 0.05902002
# [15] 0.13065495 0.13065495 0.13065495

#nuvec1
# [1] 0.04249967 0.04249967 0.04249967 0.04249967 0.04249967 0.04431920 0.03425722
# [8] 0.03463310 0.03463310 0.03463310 0.03463310 0.03463310 0.03300597 0.03300597 0.03333333
# [15] 0.03333333 0.03382827 0.03382827 0.04136801 0.04229269 0.05992018 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,

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

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)

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.”
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>

References


Examples

```r
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))
mu<-(c(0.3774461, 0.1042739, 0.09649724, 0.09649724, 0.08872055, 0.08872055, 0.0739222, 0.0739222)
nu<-(c(0.1013718, 0.1013718, 0.1013718, 0.1013718, 0.1013718, 0.1013718, 0.1095413, 0.1287447, 0.1534831))
mean<-.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),
1)
```
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,w,v,μ0,ν0,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

*Topic nonparametric
  e12.cen.EMm, 1
  e12.cen.EMS, 4
  e12.test.wtm, 7
  e12.test.wts, 10
  e12.cen.EMm, 1
  e12.cen.EMS, 4
  e12.test.wtm, 7
  e12.test.wts, 10