# Package ‘emplik2’

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.



**NeedsCompilation** no

**Repository** CRAN

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

## R topics documented:



---

`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:

$$H_o : E(g(x, y) - mean) = 0$$

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 $H_o$ is also calculated, based on the assumption that -2*log(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 $H_o$. 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

`el2.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 `el2.test.wtm` function (which is called by `el2.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 - pchisq(-2LLR, 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

Barton, W. (2010). Comparison of two samples by a nonparametric likelihood-ratio test. PhD dissertation at University of Kentucky.

Chang, M. and Yang, G. (1987). "Strong Consistency of a Nonparametric Estimator of the Survival Function with Doubly Censored Data." `Ann. Stat.`,15, pp. 1536-1547.

Dempster, A., Laird, N., and Rubin, D. (1977). "Maximum Likelihood from Incomplete Data via the EM Algorithm." `J. Roy. Statist. Soc.`, Series B, 39, pp.1-38.

Gomez, G., Julia, O., and Utzet, F. (1992). "Survival Analysis for Left-Censored Data." In Klein, J. and Goel, P. (ed.), `Survival Analysis: State of the Art.` Kluwer Academic Publishers, Boston, pp. 269-288.

Li, G. (1995). "Nonparametric Likelihood Ratio Estimation of Probabilities for Truncated Data." `J. Amer. Statist. Assoc.`, 90, pp. 997-1003.

Owen, A.B. (2001). `Empirical Likelihood`. Chapman and Hall/CRC, Boca Raton, pp. 223-227.

Turnbull, B. (1976). "The Empirical Distribution Function with Arbitrarily Grouped, Censored and Truncated Data." `J. Roy. Statist. Soc.`, Series B, 38, pp. 290-295.

Zhou, M. (2005). "Empirical likelihood ratio with arbitrarily censored/truncated data by EM algorithm." `J. Comput. Graph. Stat.`, 14, pp. 643-656.

Zhou, M. (2009) `emplik` package on CRAN website. The function `el2.cen.EMm` here extends `el.cen.EM2` inside emplik from one-sample to two-samples.

## Examples

```
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, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0)
y<-c(21, 38, 39, 51, 77, 185, 240, 289, 524, 610, 612, 677, 798, 881, 899, 946, 1010,
1074, 1147, 1154, 1199, 1269, 1329, 1484, 1493, 1559, 1602, 1684, 1900, 1952)
dy<-c(1,1,1,1,1,1,2,2,1,1,1,1,1,2,1,1,1,1,1,1,0,0,1,1,0,0,1,0,0,0)
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

$$H_o : E(g(x, y) - mean) = 0$$

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 $H_o$ is also calculated, based on the assumption that -2*log(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 $H_o$. The default is `fun=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 $mean$ 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 $H_o$. 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 $E(g(x, y))$. (The NPMLE itself should always be a feasible value for $mean$.)

## 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 |
| `temp3` | 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*(logel-logel00)` |
| `Pval` | the estimated p-value for $H_o$, computed as `1 - pchisq(-2LLR, 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 $\sum_{i=1}^{n} \sum_{j=1}^{m} (g(x_i, y_j) - mean)\mu_i \nu_j$, 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 `constraint` should be close to 0. |

### Author(s)

William H. Barton <bbarton@lexmark.com>

### References

Barton, W. (2010). Comparison of two samples by a nonparametric likelihood-ratio test. PhD dissertation at University of Kentucky.

Chang, M. and Yang, G. (1987). "Strong Consistency of a Nonparametric Estimator of the Survival Function with Doubly Censored Data." `Ann. Stat.`,15, pp. 1536-1547.

Dempster, A., Laird, N., and Rubin, D. (1977). "Maximum Likelihood from Incomplete Data via the EM Algorithm." `J. Roy. Statist. Soc.`, Series B, 39, pp.1-38.

Gomez, G., Julia, O., and Utzet, F. (1992). "Survival Analysis for Left-Censored Data." In Klein, J. and Goel, P. (ed.), `Survival Analysis: State of the Art.` Kluwer Academic Publishers, Boston, pp. 269-288.

Li, G. (1995). "Nonparametric Likelihood Ratio Estimation of Probabilities for Truncated Data." `J. Amer. Statist. Assoc.`, 90, pp. 997-1003.

Owen, A.B. (2001). `Empirical Likelihood.` Chapman and Hall/CRC, Boca Raton, pp.223-227.

Turnbull, B. (1976). "The Empirical Distribution Function with Arbitrarily Grouped, Censored and Truncated Data." `J. Roy. Statist. Soc.`, Series B, 38, pp. 290-295.

Zhou, M. (2005). "Empirical likelihood ratio with arbitrarily censored/truncated data by EM algorithm." `J. Comput. Graph. Stat.`, 14, pp. 643-656.

Zhou, M. (2009) `emplik` package on CRAN website. The `el2.cen.EMs` function extends `el.cen.EM` function from one-sample to two-samples.

### Examples

```
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(1,2,1,1,1,1,1,2,1,1,1,1,1,1,1,0,0,1,0,0,0,0,1,1,0)
y<-c(21,38,39,51,77,185,240,289,524,610,612,677,798,881,899,946,
1010,1074,1147,1154,1199,1269,1329,1484,1493,1559,1602,1684,1900,1952)
dy<-c(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,0,0,0,0,0,1,0,0,0)

# 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,

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

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.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 $g_k(xd1_i, yd1_j)$; otherwise there may be no distributions for $xd1$ and $yd1$ 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 $E(g(xd1, yd1))$. (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 algorith m |
| `lam` | the vector of Lagrangian mulipliers 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

Owen, A.B. (2001). `Empirical Likelihood`. Chapman and Hall/CRC, Boca Raton, pp.223-227.

## 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(10,85,209,273,279,324,391,566,852,881,895,954,1101,1393,1669,1823,1941)
nx1=length(xd1)
yd1<-c(21,38,39,51,77,185,524,610,612,677,798,899,946,1010,1074,1147,1154,1329,1484,1602,1952)
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, 1.261740, 2.912753,
 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, 0.03389263, 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, 1.080832,
 1.080832, 1.080832, 1.080832, 1.000000, 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, 0.04316922, 0.03425722,
 0.03463312, 0.03463312, 0.03463312, 0.03463312, 0.03300598, 0.03300598, 0.03333333,
 0.03333333, 0.03382827, 0.03382827, 0.04136800, 0.04229270, 0.05992020, 0.22762676)


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]<-(xd1[i]>yd1[j])
        H2u[i,j]<-(xd1[i]>1060) } }
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[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
# [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 0.04316920 0.03425722
# [8] 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
```

```
#        [,1]      [,2]
#  [1,] 8.9549 -10.29119
```

---

el2.test.wts — *Computes maximium-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,

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

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 $[g(u_i, v_j) - mean]$ 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 multipler found by the calculations |

### Author(s)

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

### References

Owen, A.B. (2001). `Empirical Likelihood`. Chapman and Hall/CRC, Boca Raton, pp.223-227.

### Examples

```
u<-c(10, 209, 273, 279, 324, 391, 566, 785)
v<-c(21, 38, 39, 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, 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, 0.1095413, 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))
el2.test.wts(u,v,wu,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