

Package ‘survSpearman’

July 23, 2025

Title Nonparametric Spearman's Correlation for Survival Data

Version 1.0.1

Maintainer Svetlana Eden <svetlana.k.eden.1@vumc.org>

Description Nonparametric estimation of Spearman's rank correlation with bivariate survival (right-censored) data as described in Eden, S.K., Li, C., Shepherd B.E. (2021), Nonparametric Estimation of Spearman's Rank Correlation with Bivariate Survival Data, Biometrics (under revision). The package also provides functions that visualize bivariate survival data and bivariate probability mass function.

Depends R (>= 3.5)

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Svetlana Eden [aut, cre],
Chun Li [aut, anl],
Bryan Shepherd [aut, anl]

Repository CRAN

Date/Publication 2022-09-26 17:40:02 UTC

Contents

CCASAnetData	2
data123	3
fillUpDecimals	3
fillUpStr	4
plotMatrix	5
plotVector	7
survDabrowska	8

survPMF	9
survPMFPlot	11
survRestricted	13
survSpearman	14
visualBivarTimeToEvent	16
Index	18

CCASAnetData	<i>CCASAnetData</i>
--------------	---------------------

Description

Dataset with bivariate survival times to events and times to censoring. This de-identified dataset was made available with permission from the Caribbean, Central, and South America network for HIV Epidemiology (CCASAnet). The US National Institutes of Health partially funded the data collection (grants R01 AI093234 and U01 AI069923). The dataset includes 6691 HIV-positive adults (one record per patient) from Brazil, Chile, Honduras, Mexico, and Peru sites, anonymized for presentation. Each record contains time (in years) from antiretroviral therapy (ART) initiation to viral failure, viral failure event indicator, time (in years) from ART initiation to regimen change, and regimen change indicator.

Usage

CCASAnetData

Format

- A data frame with 6691 rows and 5 variables:
- site** Anonymized study site.
 - timeToRegimenChange** Time (in years) from antiretroviral therapy initiation to regimen change.
 - regimenChange** Regimen change event indicator: 0 - no regimen change, 1 - regimen was changed.
 - timeToViralFailure** Time (in years) from antiretroviral therapy initiation to viral failure.
 - viralFailure** Viral failure event indicator: 0 - no viral failure, 1 - viral failure.

Source

Caribbean, Central, and South America network for HIV Epidemiology, CCASAnet (grants R01 AI093234 and U01 AI069923).

data123	<i>data123</i>
---------	----------------

Description

Bivariate survival data simulated from Frank’s copula family with parameter 4.426265, which is equivalent to Spearman’s correlation of 0.6. The marginal distributions of time to event and time to censoring are exponential with mean 1. In the first 100 rows, the follow-up time is restricted to time 2, and the observations can also be independently censored before time 2. In rows 101:200, the follow-up time is not restricted, but some observations are independently censored. In rows 201:300, there is no censoring.

Usage

data123

Format

- A data frame with 300 rows and 4 variables:
- X** Time to event X.
 - deltaX** Event indicator for event X (1 - event; 0 - censoring event).
 - Y** Time to event Y
 - deltaY** Event indicator for event Y (1 - event; 0 - censoring event).

Source

Simulated data.

fillUpDecimals	<i>Formats decimal part of numbers.</i>
----------------	---

Description

The function adds zeros after the decimal point of each element of a numeric vector so that all elements have the same number of digits after the decimal point.

Usage

fillUpDecimals(numVec, howManyToFill, fill)

Arguments

numVec	A vector of numbers in a numeric or character format.
howManyToFill	The target number of digits after the decimal point. If howManyToFill is NULL (the default), then the function uses the maximum number of digits after the decimal point among the element of numVec.
fill	What character to insert as a filler to reach the target number of digits (characters) after the decimal point. The default is '0'.

Details

The function adds zeros (or other characters) after the decimal point of each element of numVec so that all elements have the same number of decimal places after the decimal point. Lower and upper case letters are treated as digits (see the examples). NA values are returned unchanged. It is not recommended to include elements with characters other than digits and letters into numVec because the function will not work as described for these elements.

Value

A character string with equal number of decimal places after the decimal point.

Author(s)

Svetlana K Eden, <svetlanaeden@gmail.com>

Examples

```
fillUpDecimals(c("2", "3.4", "A", NA))
fillUpDecimals(c("2", "3.4", "A.5", NA), howManyToFill = 3)
fillUpDecimals(c("2", "3.4", "A", NA), fill = "X")
fillUpDecimals(c("2", "3.4", "A", NA), howManyToFill = -3)
fillUpDecimals(c("2", "3.4", "A.zx", NA), fill = "?")
### It is not recommended to include elements
### with characters other than digits and letters
### because the function will not work as described
### for these elements
fillUpDecimals(c("2", "3.4", "A.zx", NA, "#", "#a"), fill = "?")
```

fillUpStr	<i>Adds characters to the head or tail of each element of the character vector.</i>
-----------	---

Description

Adds characters to the head or tail of each element of the character vector to make all elements the same length.

Usage

```
fillUpStr(strVec, howManyToFill, where, fill)
```

Arguments

<code>strVec</code>	A vector of character strings.
<code>howManyToFill</code>	The target number of characters in each element of <code>strVec</code> . If <code>howManyToFill</code> is NULL or less than the number of characters in the longest element of <code>strVec</code> , then the number of characters in the longest element is used.
<code>where</code>	Where to place the characters: at the beginning of each element ('head', the default) or at the end of each element ('tail')
<code>fill</code>	What character to add. The default is a blank space.

Details

The function adds characters to the head or tail of each element of the character vector, `strVec`, so that all elements have the same number of characters. NA values are returned unchanged. If `howManyToFill` is NULL or less than the number of characters in the longest element of `strVec`, then the number of characters in the longest element is used.

Value

A character string with equal number of characters in each element.

Author(s)

Svetlana K Eden, <svetlanaeden@gmail.com>

Examples

```
fillUpStr(c("A", "aaa", NA, "bb", "4.5"), where="tail", howManyToFill = 4, fill = "_")
fillUpStr(c("A", "aaa", NA, "bb", "4.5"), where="tail", howManyToFill = -3, fill = "_")
fillUpStr(c("A", "aaa", NA, "bb", "4.5"), where="head", howManyToFill = -3, fill = "*")
```

plotMatrix

Plots a color matrix.

Description

Plots a matrix of colors at a given point. Each element of the color matrix is plotted as a rectangle with user specified side lengths and border color.

Usage

```
plotMatrix(colorMatr, borderCol, coordX, coordY, widthX, widthY)
```

Arguments

colorMatr	A matrix of colors in R format.
borderCol	The border color of plotted rectangles.
coordX	X coordinate of the lower left corner of colorMatr.
coordY	Y coordinate of the lower left corner of colorMatr.
widthX	The vector of lengths of the rectangle sides parallel to the X-axis. The length of widthX can be either 1 (all rectangles have the same side length for X-axis) or the number of columns of colorMatr
widthY	The vector of lengths of the rectangle sides parallel to the Y-axis. The length of widthY can be either 1 (all rectangles have the same side length for Y-axis) or the number of rows of colorMatr

Details

R-function `plot()` has to be called first. The function plots a color matrix at a given coordinate. Each element of the color matrix is plotted as a rectangle with user specified border color and side lengths on X- and Y-axes. Element `colorMatr[nrow(colMatr), 1]` is displayed as the bottom left rectangle. Element `colorMatr[1, ncol(colMatr)]` is displayed as the top right rectangle.

Value

None

Author(s)

Svetlana K Eden, <svetlanaeden@gmail.com>

Examples

```
### Uneven rectangles
colorMatr = matrix(c("goldenrod1", "mediumpurple3", "palegreen3",
  "royalblue1", "orchid", "firebrick1"), nrow = 2, byrow = TRUE)
plot(c(1, 4), c(2, 4), type = "n", xlab = "", ylab = "")
plotMatrix(colorMatr, borderCol = "white", coordX = 1, coordY = 2,
  widthX = c(1/4, 1, 1/6), widthY = c(1, 1/2))

### Plotting the legend:
reshapeColMatr = matrix(t(colorMatr), ncol = 1, nrow = nrow(colorMatr)*ncol(colorMatr),
  byrow = TRUE)
plotMatrix(reshapeColMatr, borderCol = "white", coordX = 2.8, coordY = 2,
  widthX = c(1/4), widthY = c(1/4))
text(x = rep(3.03, nrow(reshapeColMatr)),
  y = 2.12+c(0,cumsum(rep(1/4, nrow(reshapeColMatr)-1)))),
  labels = reshapeColMatr[nrow(reshapeColMatr):1], pos = 4, cex = 0.7)

### Same length and width rectangles on the black background
plot(c(1, 4), c(2, 4), type = "n", xlab = "", ylab = "")
polygon(x = c(0, 5, 5, 0, 0), y = c(0, 0, 5, 5, 0), col = "black")
plotMatrix(colorMatr, borderCol = "black", coordX = 1, coordY = 2, widthX = 1, widthY = 1)
```

plotVector	<i>Plots a set of rectangles that represent a numeric vector.</i>
------------	---

Description

Plots a vector of numeric values as a set of rectangles with user specified height, width, color, and border color. The bases of the rectangles are aligned like in a histogram.

Usage

```
plotVector(vectValues, width, coordX, coordY, rotationRadians, vectColor, bordColor)
```

Arguments

vectValues	The heights of the rectangles. If <code>vectValues[i]>=0</code> then the base of rectangle <code>i</code> is plotted at <code>coordY</code> and the top is plotted at <code>coordY + (abs(vectValues[i]))</code> . If the <code>vectValues[i]<0</code> then the base of rectangle <code>i</code> is plotted at <code>coordY</code> and the bottom is plotted at <code>coordY - abs(vectValues[i])</code> .
width	The width (positive number) of the rectangles. It can be either one number or a vector of numbers with the same length as <code>vectValues</code> .
coordX	X coordinate of the left base corner of the first rectangle, <code>vectValues[1]</code> .
coordY	Y coordinate of the left base corner of the first rectangle, <code>vectValues[1]</code> .
rotationRadians	By how much (in radians) to rotate the aligned rectangles around (<code>coordX</code> , <code>coordY</code>). The rotation is performed counter clock-wise.
vectColor	The colors of the rectangles in R color format. It can be either one value or a vector of values with the same length as <code>vectValues</code> .
bordColor	The color of the border of the rectangles. It can be either one value or a vector of values with the same length as <code>vectValues</code> .

Details

R-function `plot` has to be called first. The rectangles are aligned in the following manner. If `rotationRadians` is zero, then the base of all rectangles is at `coordY`. If `vectValues[i]>=0` then the top of the rectangle is plotted at `coordY + abs(vectValues[i])`. If the `vectValues[i]<0` then the bottom of the rectangle is plotted at `coordY - abs(vectValues[i])`. If `rotationRadians` is not zero, then the aligned rectangles are rotated counter clock-wise around point (`coordX`,`coordY`).

Value

None

Author(s)

Svetlana K Eden, <svetlanaeden@gmail.com>

Examples

```
### Histogram-like plot
plot(c(-1, 2), c(-1, 1), type = "n")
vectValues = c(0.057, 0.336, 0.260, 0.362, 0.209)
plotVector(vectValues, width = 0.2, coordX = 0, coordY = 0, rotationRadians = 0,
  vectColor = "gray", bordColor = "white")

### Rotated and flipped sequence of rectangles.
width = c(0.10, 0.20, 0.10, 0.80, 0.12)
vectColor = c("orange", "green", "orchid", "blue", "goldenrod1")
plot(c(-1, 2), c(-1, 2), type = "n")
plotVector(vectValues = vectValues, width, coordX = 0, coordY = 0,
  rotationRadians = pi/2, vectColor, bordColor = "white")
plotVector(vectValues = -vectValues, width, coordX = 0, coordY = 0,
  rotationRadians = 0, vectColor, bordColor = "white")

### Histogram-like plot with positive and negative values.
vectValues = c(0.057, -0.336, 0.260, -0.222, 0.209)
plot(c(-1, 1), c(-1, 1), type = "n")
vectColor = rep("goldenrod1", length(vectValues))
vectColor[vectValues<0] = "royalblue1"
bordColor = rep("red", length(vectValues))
bordColor[vectValues<0] = "darkblue"
plotVector(vectValues, width = 0.4, coordX = -1, coordY = 0, rotationRadians = 0,
  vectColor, bordColor = bordColor)
```

survDabrowska

Computes marginal and joint survival probability estimates using Dabrowska's method.

Description

The function computes marginal and joint survival probabilities for right-censored data using Dabrowska's (1988) method.

Usage

```
survDabrowska(X, Y, deltaX, deltaY)
```

Arguments

X	Time to event or censoring for variable X. It indicates time to event if argument deltaX is 1 and time to censoring if argument deltaX is 0.
Y	Time to event or censoring for variable Y. It indicates time to event if argument deltaY is 1 and time to censoring if argument deltaY is 0.
deltaX	Event indicator for variable X. deltaX is 1 if the event is observed and 0 if it is censored.
deltaY	Event indicator for variable Y. deltaY is 1 if the event is observed and 0 if it is censored.

Details

The function returns a list with two values: Dabrowska's estimator of the marginal and joint survival probabilities, `DabrowskaEst`, and its corresponding marginal and joint cumulative distribution functions (CDFs), `DabrowskaCDF`, based on the bivariate survival data provided by the user.

Value

A list of two elements: `DabrowskaEst` and `DabrowskaCDF`. `DabrowskaEst` is a matrix containing marginal and joint survival probabilities. The first column is the marginal survival probability corresponding to variable X . The first row is the marginal survival probability corresponding to variable Y . The rest of the matrix contains the joint survival probabilities. The row names of `DabrowskaEst` are ordered X -values. The column names of `DabrowskaEst` are ordered Y -values. Element `DabrowskaEst[1,1]` equals 1. Its row and column name is `'0'`. `DabrowskaCDF` is a matrix containing marginal and joint cumulative distribution functions (CDFs). The first row of `DabrowskaCDF` is the marginal CDF corresponding to variable X . The first column of `DabrowskaCDF` is the marginal CDF corresponding to variable Y . The row and column names of `DabrowskaCDF` are the same as for `DabrowskaEst`.

Author(s)

Svetlana K Eden, <svetlanaeden@gmail.com>

References

Dabrowska, D. M. (1988). Kaplan–Meier estimate on the plane. *The Annals of Statistics* 16, 1475–1489.

Examples

```
X = c(0.5, 0.6, 0.6, 0.8)
Y = c(0.44, 0.77, 0.88, 0.99)
deltaX = c(1, 0, 1, 1)
deltaY = c(1, 1, 1, 1)
survDabrowska(X, Y, deltaX, deltaY)
```

survPMF	<i>Computes marginal and joint probability mass function from marginal and joint survival probabilities.</i>
---------	--

Description

The function computes marginal and joint probability mass functions from marginal and joint survival probabilities.

Usage

```
survPMF(bivarSurf)
```

Arguments

bivarSurf A matrix containing the marginal and joint survival probabilities. The first column is the marginal survival probability corresponding to variable X . The first row is the marginal survival probability corresponding to variable Y . The rest of the matrix contains the joint survival probabilities. The row names of **bivarSurf** are ordered X values. The column names of **bivarSurf** are ordered Y values. Element **bivarSurf**[1,1] equals 1. Its row and column name is '0' (see the documentation for the return value **DabrowskaEst** in function **survDabrowska**).

Details

The function returns a list of survival surfaces and their differentials. Element **Sdxdy** of this list is the marginal and joint probability mass function in the same format as argument **bivarSurf**. The rest of the returned list elements are matrices in the same format as **bivarSurf** except that they do not contain marginal values and row/column names.

Value

The following list of survival surfaces and their differentials is returned. **Sdxdy** is the marginal and joint probability mass functions in the same format as argument **bivarSurf**; **Sxy** is the joint survival probability; **SxMyM** is **Sxy** at point $(x-, y-)$, where $x-$ is the left limit of x ; **Sx** is the marginal survival probability function for variable X ; **Sy** is the marginal survival probability function for variable Y ; **Sdx** is the marginal probability mass function for variable X ; **Sdy** is the marginal probability mass function for variable Y ; **SxM** is the marginal survival probability function for X at point $x-$; **SyM** is the marginal survival probability function for Y at point $y-$; **SxM_y** is the joint survival probability function at point $(x-, y)$; **Sx_yM** is the joint survival probability function at point $(x, y-)$; **Sdx_y** is **SxM_y** - **Sxy**; **Sx_dy** is **Sx_yM** - **Sxy**; **Sdx_yM** is **SxMyM** - **Sx_yM**; **SxM_dy** is **SxMyM** - **SxM_y**.

Author(s)

Svetlana K Eden, <svetlanaeden@gmail.com>

Examples

```
X = c(0.5, 0.6, 0.8)
Y = c(0.44, 0.77, 0.99)
deltaX = c(1, 0, 1)
deltaY = c(1, 1, 1)

bivarSurf = survDabrowska(X, Y, deltaX, deltaY)$DabrowskaEst
bivarSurf

bivarPMF = survPMF(bivarSurf)$Sdxdy
bivarPMF
```

survPMFPlot

*Plots marginal and joint probability mass functions.***Description**

Plots marginal and joint probability mass functions for bivariate survival data.

Usage

```
survPMFPlot(bivarSurf, gridWidthX, gridWidthY,
scaleGapX, scaleGapY, scaleHistX, scaleHistY,
XAxisLabel, YAxisLabel, lineXAxisLabel, lineYAxisLabel,
timeLabelScale, axisLabelScale,
labelSkipX, labelSkipY, roundX, roundY,
plotLabel, linePlotLabel)
```

Arguments

bivarSurf	A matrix containing marginal and joint survival probabilities. The first column is the marginal survival probability corresponding to variable X. The first row is the marginal survival probability corresponding to variable Y. The rest of the matrix contains the joint survival probabilities. The row names of bivarSurf are ordered X values. The column names of bivarSurf are ordered Y values. Element bivarSurf[1,1] equals 1. Its row and column name is '0' (see the documentation for the return value DabrowskaEst in function survDabrowska)
gridWidthX	Grid size on X-axis.
gridWidthY	Grid size on Y-axis.
scaleGapX	The proportion by which the gap between the marginal and joint histograms along the X-axis is increased (if >1) or decreased (if >0 and <1).
scaleGapY	The proportion by which the gap between the marginal and joint histograms along the Y-axis is increased (if >1) or decreased (if >0 and <1).
scaleHistX	The proportion by which to increase (if >1) or decrease (if >0 and <1) the height of the marginal histogram along the X-axis.
scaleHistY	The proportion by which to increase (if >1) or decrease (if >0 and <1) the height of the marginal histogram along the Y-axis.
XAxisLabel	X-axis label.
YAxisLabel	Y-axis label.
lineXAxisLabel	Line where to place the X-axis label (used by function mtext).
lineYAxisLabel	Line where to place the Y-axis label (used by function mtext).
timeLabelScale	The proportion by which the axis labels is increased (if >1) or decreased (if >0 and <1).
axisLabelScale	The proportion by which the time grid labels are increased (if >1) or decreased (if >0 and <1).

labelSkipX	If the time grid looks too busy, not all X-label time grid labels have to be printed. For example, if labelSkipX = 1 then every other label is printed starting from the first (time 0). If labelSkipX = 2 then the 1st, 4rd, 7th, 10th, ... and so on, labels are printed. The default is 0, which means that all time grid labels are printed.
labelSkipY	If the time grid looks too busy, not all Y-label time grid labels have to be printed. For example, if labelSkipY = 1 then every other label is printed starting from the first (time 0). If labelSkipY = 2 then the 1st, 4rd, 7th, 10th, ... and so on, labels are printed. The default is 0, which means that all time grid labels are printed.
roundX	Number of decimal places in time grid labels for X-label.
roundY	Number of decimal places in time grid labels for Y-label.
plotLabel	Plot label.
linePlotLabel	Line where to place the plot label (used by function mtext).

Details

Plots marginal and joint probability mass functions (PMFs) from marginal and joint survival probabilities. The probability mass gets aggregated into cells according to the user-specified arguments `gridWidthX` and `gridWidthY`. After this aggregation, the negative values (if any) are set to zero. Marginal probability mass functions are displayed as histograms. Joint probability mass function is displayed as a matrix with darker cells indicating larger probability mass aggregated in this cell. Zero mass is denoted with a very faint gray shade. Because the shading is relative, with greater range of probability mass more gray shades are observed. A future version of the function will allow users to choose their own shading/color function.

Value

None

Author(s)

Svetlana K Eden, <svetlanaeden@gmail.com>

References

Eden, S.K., Li, C., Shepherd B.E. (2021). Non-parametric Estimation of Spearman's Rank Correlation with Bivariate Survival Data. Biometrics (under revision).

Examples

```
X = c(0.5, 0.61, 0.6, 0.8, 0.78, 0.7, 0.9)
Y = c(0.44, 0.15, 0.77, 0.88, 0.22, 0.99, .33)
deltaX = c(1, 0, 1, 1, 0, 1, 0)
deltaY = c(1, 0, 1, 0, 1, 1, 1)

dabrSurf = survDabrowska(X, Y, deltaX, deltaY)$DabrowskaEst

grid = 0.1
```

```

survPMFPlot(bivarSurf = dabrSurf, gridWidthX = grid, gridWidthY = grid,
            scaleGapX = 1, scaleGapY = 1,
            XAxisLabel = "X", YAxisLabel = "Y", timeLabelScale = 1,
            axisLabelScale = 1,
            labelSkipX = 0, labelSkipY = 0, roundX = 2, roundY = 2,
            plotLabel = "Bivariate PMF")

```

survRestricted	<i>Computes conditional marginal and joint survival probability in a restricted region.</i>
----------------	---

Description

The function computes marginal and joint survival probabilities conditionally on surviving in a restricted region. This region is defined by the user as $[0, \tau_X) \times [0, \tau_Y)$.

Usage

```
survRestricted(bivarSurf, tauX = Inf, tauY = Inf)
```

Arguments

bivarSurf	A matrix containing marginal and joint survival probabilities. The first column is the marginal survival probability corresponding to variable X. The first row is the marginal survival probability corresponding to variable Y. The rest of the matrix contains the joint survival probabilities. The row names of bivarSurf are ordered X values. The column names of bivarSurf are ordered Y values. Element <code>bivarSurf[1,1]</code> equals 1. Its row and column name is '0' (see the documentation for the return value <code>DabrowskaEst</code> in function <code>survDabrowska</code>).
tauX	The X value that defines the restricted region for the X variable.
tauY	The Y value that defines the restricted region for the Y variable.

Details

The method of Dabrowska can result in negative probability mass for some points, which may result in zero or negative probability of failure in the restricted region. This only happens when the sample size is small and censoring is heavy. If the probability of survival in the restricted region is zero or less, NA value is returned. Otherwise, the function returns a list of survival probabilities and their differentials conditionally on being in the restricted region defined by `tauX` and `tauY`. Element `Sxy` of this list is the conditional marginal and joint survival probabilities with row/column names in the same format as argument `bivarSurf`. The rest of the returned list elements are matrices in the same format as `bivarSurf` except that they do not contain marginal values and row/column names.

Value

The function returns the following list of survival surfaces and their differentials: S_{xy} is the conditional marginal and joint survival probabilities in the same format as `bivarSurf`; S_{xMyM} is S_{xy} at point $(x-, y-)$, where $x-$ is the left limit of x ; S_x is the conditional marginal survival probability function for variable X ; S_y is the conditional marginal survival probability function for variable Y ; S_{dx} is the conditional marginal probability mass function for variable X ; S_{dy} is the conditional marginal probability mass function for variable Y ; S_{xM} is the conditional marginal survival probability function for X at point $x-$; S_{yM} is the conditional marginal survival probability function for Y at point $y-$; S_{xM_y} is the conditional joint survival probability function at point $(x-, y)$; S_{x_yM} is the conditional joint survival probability function at point $(x, y-)$; S_{dx_y} is $S_{xM_y} - S_{xy}$; S_{x_dy} is $S_{x_yM} - S_{xy}$; S_{dx_yM} is $S_{xMyM} - S_{x_yM}$; S_{xM_dy} is $S_{xMyM} - S_{xM_y}$; S_{dxdy} is the conditional joint probability mass function.

Author(s)

Svetlana K Eden, <svetlanaeden@gmail.com>

References

Eden, S.K., Li, C., Shepherd B.E. (2021). Non-parametric Estimation of Spearman's Rank Correlation with Bivariate Survival Data, *Biometrics* (under revision).

Examples

```
X = c(0.5, 0.6, 0.8)
Y = c(0.44, 0.77, 0.99)
deltaX = c(1, 0, 1)
deltaY = c(1, 1, 1)
bivarSurf = survDabrowska(X, Y, deltaX, deltaY)$DabrowskaEst
bivarSurf

condSurf = survRestricted(bivarSurf, tauX = Inf, tauY = 0.88)$Sxy
condSurf
```

survSpearman

Computes Spearman's Correlation for Bivariate Survival Data.

Description

Computes non-parametric estimates of Spearman's rank correlation for bivariate survival data. Two correlations are returned: a highest rank correlation that can be interpreted as Spearman's correlation after assigning a highest rank to observations beyond a specified region, and a restricted correlation that estimates Spearman's correlation within the specified region.

Usage

```
survSpearman(X = NULL, Y = NULL, deltaX = NULL, deltaY = NULL, data = NULL,
tauX = Inf, tauY = Inf, bivarSurf = NULL)
```

Arguments

X	Time to event or censoring for variable X. It indicates time to event if argument <code>deltaX=1</code> and time to censoring if argument <code>deltaX=0</code> .
Y	Time to event or censoring for variable Y. It indicates time to event if argument <code>deltaY=1</code> and time to censoring if argument <code>deltaY=0</code> .
deltaX	Event indicator for variable X. <code>deltaX=1</code> if the event is observed and 0 if it is censored.
deltaY	Event indicator for variable Y. <code>deltaY=1</code> if the event is observed and 0 if it is censored.
data	Data frame containing variables (arguments) X, Y, <code>deltaX</code> , and <code>deltaY</code> .
tauX	The X value that defines the restricted region for X variable.
tauY	The Y value that defines the restricted region for Y variable.
bivarSurf	A matrix containing the marginal and joint survival probabilities. The first column is the marginal survival probability corresponding to variable X. The first row is the marginal survival probability corresponding to variable Y. The rest of the matrix contains the joint survival probabilities. The row names of <code>bivarSurf</code> are ordered X-values. The column names of <code>bivarSurf</code> are ordered Y-values. Element <code>bivarSurf[1,1]</code> equals 1. Its row and column name is '0' (see the documentation for the return value <code>DabrowskaEst</code> in function <code>survDabrowska</code>).

Details

The function computes the highest rank and restricted Spearman's correlations with bivariate survival data. The data can be supplied in three ways: 1) as vectors X, Y, `deltaX`, and `deltaY`; 2) as data frame data that contains the variables mentioned in 1); and 3) as matrix `bivarSurf` containing marginal and joint survival probabilities. If `bivarSurf` is not NULL then 1) and 2) are ignored. If `bivarSurf` is NULL and data is not then 2) is used. If `bivarSurf` and data are NULL then 1) is used. The highest rank correlation is the Spearman's correlation that can be interpreted as Spearman's rank correlation computed after assigning the highest rank to the events outside of `tauX` and `tauY`. The restricted Spearman's correlation is Spearman's correlation computed within the restricted region defined by `tauX` and `tauY`. Note that given `tauX` and `tauY` the survival probability is estimated using the values that are just above the latest observed event times within that region, what we call an effective restricted region. This means that if, for example, `tauX` is greater than the latest observed event time for X variable and `tauY` is greater than the latest observed event time for Y variable, then `tauX` and `tauY` do not affect the correlation values since the effective restricted region remains the same (as defined by the maximum observed event or censoring event times). The method of Dabrowska can result in negative probability mass for some points. This may result in zero or negative probability of failure in the restricted region, in which case the restricted Spearman's correlation cannot be computed and NA value is returned. This only happens when the sample size is small and censoring is heavy.

Value

The function returns the following list of values. 'Restricted region set by user' is a character vector of two user-specified restricted region values, `tauX` and `tauY`. 'Effective restricted

region' is character vector of two values that define the effective restricted region, the values that are just above the latest observed event times within the user-specified restricted region. 'Correlation' is a numeric vector of two correlation values: the highest rank Spearman's correlation ('HighestRank') and the restricted region Spearman's correlation ('Restricted'), where the restricted region is defined by the values in 'Effective restricted region'.

Author(s)

Svetlana K Eden, <svetlanaeden@gmail.com>

References

Dabrowska, D. M. (1988) Kaplan–Meier estimate on the plane. The Annals of Statistics 16, 1475–1489.

Eden, S.K., Li, C., Shepherd B.E. (2021). Non-parametric Estimation of Spearman's Rank Correlation with Bivariate Survival Data. Biometrics (under revision).

Examples

```
### Compute correlation from data
X <- c(0.5, 0.6, 0.7, 0.8)
Y <- c(0.44, 0.77, 0.88, 0.99)
deltaX <- c(1, 0, 1, 1)
deltaY <- c(1, 1, 1, 1)
survSpearman(X, Y, deltaX, deltaY)
survSpearman(X, Y, deltaX, deltaY, tauX = 100, tauY = 100)
survSpearman(X, Y, deltaX, deltaY, tauX = 100, tauY = 0.99)
survSpearman(X, Y, deltaX, deltaY, tauX = 0.8, tauY = 0.99)

### Compute correlation from survival surface
someSurf <- survDabrowska(X, Y, deltaX, deltaY)$DabrowskaEst
survSpearman(tauY = 0.9, bivarSurf = someSurf)
```

visualBivarTimeToEvent

Plots bivariate right-censored data.

Description

Plots bivariate right-censored data distinguishing between uncensored, singly censored, and doubly censored observations.

Usage

```
visualBivarTimeToEvent(X, Y, deltaX, deltaY, labelX, labelY,
  xlim = NULL, ylim = NULL, dotSize = 0.7,
  segLength=abs(mean(diff(c(X, Y)))), scaleLegendGap = 1,
  legendCex = 1, labCex = 1, axisCex = 1)
```


Arguments

X	Time to event or censoring for variable X. It indicates time to event if argument <code>deltaX=1</code> and time to censoring if argument <code>deltaX=0</code> .
Y	Time to event or censoring for variable Y. It indicates time to event if argument <code>deltaY=1</code> and time to censoring if argument <code>deltaY=0</code> .
deltaX	Event indicator for variable X. <code>deltaX=1</code> if the event is observed and 0 if it is censored.
deltaY	Event indicator for variable Y. <code>deltaY=1</code> if the event is observed and 0 if it is censored.
labelX	Label for the X event.
labelY	Label for the Y event.
xlim	The range for the X-axis (the same as parameter <code>xlim</code> in function <code>plot()</code>).
ylim	The range on the Y-axis (the same as parameter <code>ylim</code> in function <code>plot()</code>).
dotSize	The size of the points (the same as parameter <code>cex</code> in function <code>points()</code>).
segLength	The length of the segment representing censored observations.
scaleLegendGap	Increases (if > 1) or decreases (if < 1) the distance between the labels in the legend;
legendCex	The size of the legend font (the same as parameter <code>cex</code> in function <code>text()</code>).
labCex	The size of <code>xlab</code> and <code>ylab</code> (the same as parameter <code>cex.lab</code> in function <code>plot()</code>).
axisCex	The size of axis labels (the same as parameter <code>cex.axis</code> in function <code>plot()</code>).

Details

Plots bivariate right-censored data distinguishing between uncensored, singly censored, and doubly censored observations. The singly and doubly censored observations are plotted as diamonds and squares, respectively, with a short segment on the right, which length is the same for X and Y. The legend is always plotted in the left upper corner.

Value

None

Author(s)

Svetlana K Eden, <svetlanaeden@gmail.com>

Examples

```
X = c(0.5, 0.61, 0.6, 0.8, 0.78, 0.7, 0.9)
Y = c(0.44, 0.15, 0.77, 0.88, 0.22, 0.99, .33)
deltaX = c(1, 0, 1, 1, 0, 1, 0)
deltaY = c(1, 0, 1, 0, 1, 1, 1)

visualBivarTimeToEvent(X, Y, deltaX, deltaY, xlim = c(0, 1), ylim = c(0, 1),
  labelX = "X", labelY = "Y", segLength = 0.05)
```

Index

- * **Dabrowska**
 - survDabrowska, [8](#)
 - * **PMF**
 - survPMF, [9](#)
 - survPMFPlot, [11](#)
 - * **Spearman**
 - survSpearman, [14](#)
 - * **bivariate**
 - survDabrowska, [8](#)
 - survPMF, [9](#)
 - survPMFPlot, [11](#)
 - survRestricted, [13](#)
 - survSpearman, [14](#)
 - visualBivarTimeToEvent, [16](#)
 - * **character**
 - fillUpStr, [4](#)
 - * **color**
 - plotMatrix, [5](#)
 - plotVector, [7](#)
 - * **conditional**
 - survRestricted, [13](#)
 - * **correlation**
 - survSpearman, [14](#)
 - * **datasets**
 - CCASAnetData, [2](#)
 - data123, [3](#)
 - * **data**
 - visualBivarTimeToEvent, [16](#)
 - * **decimal**
 - fillUpDecimals, [3](#)
 - * **digits**
 - fillUpDecimals, [3](#)
 - * **histogram**
 - plotVector, [7](#)
 - * **matrix**
 - plotMatrix, [5](#)
 - * **plot**
 - survPMFPlot, [11](#)
 - visualBivarTimeToEvent, [16](#)
 - * **probability**
 - survRestricted, [13](#)
 - * **rotated**
 - plotVector, [7](#)
 - * **survival**
 - survDabrowska, [8](#)
 - survPMF, [9](#)
 - survRestricted, [13](#)
 - survSpearman, [14](#)
 - * **vector**
 - fillUpStr, [4](#)
- CCASAnetData, [2](#)
- data123, [3](#)
- fillUpDecimals, [3](#)
- fillUpStr, [4](#)
- plotMatrix, [5](#)
- plotVector, [7](#)
- survDabrowska, [8](#)
- survPMF, [9](#)
- survPMFPlot, [11](#)
- survRestricted, [13](#)
- survSpearman, [14](#)
- visualBivarTimeToEvent, [16](#)