Package 'Biograph'

March 31, 2016

Type Package

Title Explore Life Histories

Version 2.0.6

Depends R (>= 3.0.0)

Imports mvna,etm,msm,Epi,lubridate,reshape,plyr,survival,mstate,ggplot2, grDevices,graphics,stats

Date 2016-03-31

Author Frans Willekens

Maintainer Frans Willekens <willekens@nidi.nl>

Description Transition rates are computed from transitions and exposures.Useful graphics and life-course indicators are computed. The package structures the data for multistate statistical and demographic modeling of life histories.

License GPL-2

LazyLoad yes

NeedsCompilation no

Repository CRAN

Date/Publication 2016-03-31 17:50:43

R topics documented:

`Biograph-package` *Explore life histories (biographies, lifepaths)*

Description

Biograph is designed to facilitate the descriptive and statistical analysis of life histories. It follows a multistate perspective on the life course and conceptualizes the life course as a sequence of states and transitions between states (events). Transitions are governed by transition rates that depend on event counts and exposure times. Transition rates also depend on covariates.

Biograph produces useful life-course indicators for individuals and groups: (1) types of episodes (open, closed) and transitions (transient, absorbing), (2) states occupied at various ages, (3) sojourn times in different states and (4) state sequences (trajectories). Biograph estimates transition rates by age, origin and destination. Biograph also structures the data for multistate statistical and demographic modeling of life histories. Biograph accepts data in wide format. The user needs to create a Biograph object from original data. The object has a fixed data structure and is input to most functions. The Doc subdirectory of the package (inst/doc of the source package) contains a description of how to prepare Biograph objects. It includes illustrations and R code. It demonstrates the preparation of Biograph objects using hypothetical data and data from the German Life History Survey (GLHS) (used by Blossfeld and Rohwer in their book Techniques of Event History Modeling, 2002), the Netherlands Family and Fertility Survey (NLOG98), the Survey on Health, Ageing and Retirement in Europe (SHARE), the National Family Life Survey of India (NFHS) and data from the European Registry for Blood and Marrow Transplantation (EBMT) (included in the mstate package developed by Putter and colleagues).

Two time scales are considered: individual time (age) and calendar time.

Details

Package:	Biograph
Type:	Package
Version:	2.0.6
Date:	2016-03-31
License:	GPL-2
LazyLoad:	yes

Major functions of Biograph (for a complete list, see the INDEX file):

Parameters : extract information from data set

TransitionAB : extract from the raw data information on a selected transition (age profile)

Occup : determines state occupancies at each age and state-specific sojourn times in age interval

Sequences : shows the different state sequences (trajectories) in the sample data

Rates.ac : computes transition rates by origin state and destination state, and by age (and covariates) by dividing occurrences and exposures.

Biograph.long : converts Biograph object to long format required by the survival and eha packages

Biograph.Lexis : converts Biograph object to Lexis object

Biograph.mstate : converts Biograpgh object to long format required by the mstate package

Biograph.msm : converts Biograph object to long format required by the msm package

Biograph.mvna : conferts Biograph object to long format required by the mvna package

Cumrates : produces (and plots) two types of estimates of cumulative transition rates: Nelson-Aalen estimators (using the mvna package) and occurrence-exposure rates. The latter are generally used in demography.

Lexispoints : produces a scatter plot of the age and time at selected transitions for a subsample of subjects with selected characteristics

Lexislines.episodes : draws lifelines for selected subjects

Author(s)

Frans Willekens <willekens@demogr.mpg.de>

References

Willekens, F. (2014) Biograph. Multistate analysis of life histories with R. Springer

---

`AgeTrans` *Ages at transition*

Description

Converts dates at transition to ages at transition

Usage

```
AgeTrans(Bdata)
```

Arguments

`Bdata`	Biograph object: data in Biograph format

Details

Ages are in years, even when dates are in CMC.

Value

ages	ages at transition
ageentry	ages at entry into observation
agecens	ages at end of observation (censoring)
st_entry	states occupied at entry into observation
st_censoring	states occupied at censoring

Note

The sequence of transitions in the component 'ages' is same as in the Biograph object

Author(s)

Frans Willekens

See Also

YearTrans

Examples

```
data(GLHS)
agetrans <- AgeTrans(Bdata=GLHS)
```

---

age_as_Date *Converts a vector of ages to object of class 'Date'.*

Description

Converts vector of ages to object of class 'Date'.

Usage

```
age_as_Date (x,born,format.born,format.out)
```

Arguments

x	vector of ages
born	vector of dates of birth
format.born	Format of birth dates
format.out	Format of output dates

Value

decimal year (year and fraction of year)

Author(s)

Frans Willekens

Examples

```
age_as_Date (23.456,"1990-10-30","%Y-%m-%d")
# date is 14 April 2014
```

`age_as_year` *Converts a vector of ages to decimal year (calendar year and fraction of year).*

Description

Converts a vector of ages (class 'numeric') to calendar years and fractions of year. The function needs dates of births.

Usage

```
age_as_year(x,born,format.born)
```

Arguments

`x`	vector of ages
`born`	vector of dates of birth (class 'Date')
`format.born`	format of birth dates

Value

`year`	Calendar year and fraction of year

Author(s)

Frans Willekens

Examples

```
age_as_year(23,"1990-10-30","%Y-%m-%d")
```

Biograph.Lexis — *Converts Biograph object to Lexis object of class "Lexis"*

Description

Converts Biograph object to Lexis object.

Usage

```
Biograph.Lexis(Bdata,Dlong)
```

Arguments

Bdata	Biograph object
Dlong	Data in episode (long) format

Details

The argument Dlong is Dlong$Depisode. If Dlong is missing, it is computed.

Value

Lexis object. Time scale is age and time unit is year.

Author(s)

Frans Willekens

Examples

```
data (GLHS)
Dlongg <- Biograph.long (GLHS)
D <- Biograph.Lexis (GLHS,Dlongg$Depisode)

D<- Biograph.Lexis (GLHS)
```

Biograph.long — *Converts Biograph object to long format*

Description

Converts Biograph object to long format used by e.g. the survival and eha packages.

Usage

```
Biograph.long(Bdata)
```

Arguments

Bdata	Biograph object

Details

The function firsts calls Parameters if that function was not called before. The long format is produced by the reshape function. A few variables are added. The long format produced by Biograph.long is used by the survival and eha packages. The mstate, mvna and msm packages require a different long format. Biograph.mstate, Biograph.mvna and Biograph.msm produce these formats.

Value

Devent	Data frame with event data: one record for each transition
Depisode	Data frame with episode data: one record for each episode

Author(s)

Frans Willekens

References

Willekens (2011)

See Also

Biograph.mstate, Biograph.mvna, Biograph.msm

Examples

```
data(GLHS)
D <- Biograph.long (Bdata=GLHS)
```

Biograph.msm *Biograph to msm format*

Description

Converts Biograph object to input data for msm package (long format)

Usage

```
Biograph.msm(Bdata)
```

Arguments

Bdata	Biography object

Value

Data frame with data in msm format

Author(s)

Frans Willekens

References

Jackson, C. (2011). Multi-state models for panel data: the msm package for R. Journal of Statistical Software, 38(8), 28 pp.

Jackson, C. (2013). msm: Multi-state Markov and hidden Markov models in continuous time. Available at the CRAN repository.

Examples

```
data(GLHS)
D <- Biograph.msm (GLHS)
# Function statetable.msm of msm package
require(msm)
statetable.msm(state,ID,data=D)
```

Biograph.mstate *Converts Biograph object to mstate object of class "msdata"*

Description

Converts Biograph object to input data for mstate package (long format).

Usage

```
Biograph.mstate(Bdata)
```

Arguments

Bdata	Biograph object

Details

The procedure consists of four steps:

a. Remove intra-state transitions. First the functions Parameters and Remove.intrastate are called.

b. Produce long format

c. Create object of class "msdata" (mstate format) from long format by producing one record of input data for each possible destination from a given origin and by adding the attribute "trans".

Value

Data in msdata format, used by the mstate package.

Author(s)

Frans Willekens

References

Putter, H., M. Fiocco and R.B. Geskus (2007) Tutorial in biostatistics: competing risks and multi-state models. Statistics in Medicine, 26:2389-2430.

De Wreede, L.C., M. Fiocco and H. Putter (2010) The mstate package for estimation and prediction in non- and semi-parametric multi-state and competing risks models. Computer Methods and Programs in Biomedicine, doi:10.1016/j.cmpb.2010.01.001

Examples

```
data (GLHS)
D <- Biograph.mstate (GLHS)
# Function events of mstate package
library (mstate)
events (D)
s <- coxph(Surv(Tstart,Tstop,status)~edu+sex,data=D,method="breslow")
```

`Biograph.mvna` *Converts Biograph object to input data for mvna package*

Description

Converts Biograph object to long format used as input data in mvna package.

Usage

```
Biograph.mvna(Bdata)
```

Arguments

`Bdata`	Biograph object

Details

The function performs four operations on data in Biograph format:

a. It checks whether intrastate transitions are omitted, i.e. that the diagonal elements are zero. If that is not the case, then it calls the Remove.intrastate function to remove the intrastate moves.

b. It calls the Parameters function to determine the parameters associated with the Biograph object with diagonal elements removed.

c. It calls the Biograph.long function to create an object having the data in long format.

d. It adds to the data frame the variables entry and exit and it changes the variable name of the subject identification number from ID to id.

Value

D	Data frame of class "mvna" (only variables that are required)
D.cov	Data in mvna format, including covariates and some other variables
par	Object produced by Parameters function, including the object trans_possible: possible transitions
cens	Character string denoting censoring (it is "cens")

Note

Function Biograph.mvna uses the functions Extract, remove.intrastate and Biograph.long

Author(s)

Frans Willekens

References

Allignol, A., J. Beyersmann and M. Schumacher (2008) mvna: An R package for the Nelson-Aalen estimator in multistate models. R Newsletter, 8(2):48-50

Allignol, A., M. Schumacher and J. Beyersmann (2011). Empirical transition matrix of multistate models: the etm package. Journal of Statistical Software, 38(4), 15 pp.

Examples

```
data(GLHS)
D <- Biograph.mvna (GLHS)
# Run mvna:
require (mvna)
zz <- attr(D$D,"param")$namstates
zzz <- attr(D$D,"param")$trans_possible
na <- mvna(data=D$D,state.names=zz,tra=zzz,cens.name=D$cens)
```

ChangeObservationWindow.e

Changes observation window to period between two transitions

Description

Observation starts at time of entry into given state (entrystate) and ends at time of entry into another given state (exitstate). If exitstate = NULL, then observation ends at last date for which data are available. Entry into entrystate and entry into exitstate must lie in original observation window.

Usage

```
ChangeObservationWindow.e(Bdata, entrystate, exitstate)
```

Arguments

Bdata	Biograph object
entrystate	Observation starts at time of entry into entry state
exitstate	Entry into exitstate implies end of observation. If exitstate = NA, observation ends at end of initial observation window

Value

Biograph object with information on new observation window and transitions during the new observation window.

Author(s)

Frans Willekens

Examples

```
#  The following example defines observation window
# as period between labour force entry and survey date.
data(GLHS)
entrystate <- "J"
exitstate <- NA
GLHS2 <- ChangeObservationWindow.e (GLHS,entrystate,exitstate)
```

ChangeObservationWindow.t

Changes observation window to period between two points in time

---

Description

Observation starts at starttime and ends at endtime.

Usage

```
ChangeObservationWindow.t(Bdata, starttime, endtime,covs.dates)
```

Arguments

Bdata	Biograph object
starttime	Observation starts at starttime. Time is measured in unit used in Biograph object.
endtime	Observation ends at endtime. Time is measured in unit used in Biograph object.
covs.dates	Covariates having dates

Value

Biograph object with information on new observation window and transitions during the new observation window.

Author(s)

Frans Willekens

Examples

```
#  The following example defines observation window
# as period between Century Month Code (CMC) 500 and CMC 900 (survey = GLHS)
data (GLHS)
z <- ChangeObservationWindow.t (GLHS,starttime=500,endtime=800,covs.dates=c("marriage","LMentry"))
```

---

check.par *Checks major characteristics of Biograph object*

Description

Assesses consistency between parameter values and the Biograph object (data set) loaded

Usage

```
check.par(Bdata)
```

Arguments

Bdata	Biograph object

Details

The function checks the presence of the date format and the correct specification of a set of parameters.

Author(s)

Frans Willekens

cmc_as_age — *Converts date in Century Month Code (CMC) to age.*

Description

Converts date in Century Month Code (CMC) to age.

Usage

```
cmc_as_age (x,born,format.born)
```

Arguments

x	Date in Century Month Code
born	Date of birth
format.born	Format of date of birth

Value

year	Dates in calendar year and fraction of year
age	age

Author(s)

Frans Willekens

Examples

```
cmc_as_age (1260,555,"cmc")
cmc_as_age (1260,"1946-03-10","%Y-%m-%d")
```

cmc_as_Date — *Converts a vector of dates in Century Month Code (CMC) to object of class 'Date'.*

Description

Converts vector of dates in Century Month Code (CMC) to object of class 'Date'.

Usage

```
cmc_as_Date (x,selectday,format.out)
```

Arguments

x	Date in Century Month Code
selectday	If input date is in Century Month Code (year and month), selectday gives the day of the month. Default value is one, the transition occurs on the 1st of the month.
format.out	Format of output dates

Value

Dates (object of class 'Date'): days since 1-1-1970 and printed as date in Gregorian calendar, e.g. "1946-03-01"

Author(s)

Frans Willekens

Examples

```
# CMC 555, which is March 1,1946 or 1946.167
cmc_as_Date (x=555,selectday=1)
```

---

cmc_as_year *Converts date in Century Month Code (CMC) to date in calendar year and fraction of year.*

Description

Converts date in Century Month Code (CMC) to date in year and fraction of year.

Usage

```
cmc_as_year (x,selectday)
```

Arguments

x	Date in Century Month Code
selectday	If input date is in Century Month Code, selectday gives the day of the month. Default value is one, the transition occurs on the 1st of the month.

Value

Dates in calendar year and fraction of year

Author(s)

Frans Willekens

Examples

```
# CMC 555, which is March 1,1946 or 1946.167
cmc_as_year (x=555,selectday=1)
```

`Cumrates` *Estimates cumulative transition rates*

Description

Estimates cumulative transition rates by origin, destination, and age. Two estimation methods are distinguished: the statistical method (Nelson-Aalen estimator) and the demographic method (occurrence-exposure rates).

Usage

```
Cumrates(irate, Bdata)
```

Arguments

`irate`	Indicator variable defining option to be used: 1 = Nelson-Aalen method; 2 = Occurrence-Exposure rate; 3 = Both
`Bdata`	Biograph object

Details

The Nelson- Aalen estimator is obtained using the mvna package. First the function Biograph.mvna is called to convert the Biograph object to a data frame of class mvna. Then the mvna and the predict functions of the mvna package are called. The predict function produces cumulative hazards (with confidence intervals) at each birthday.

For estimating the cumulative occurrence-exposure rates, the functions Occup, Trans and RateTable are called (among other functions). The object M is produced; it is a three-dimensional array containing the transition rates by age, origin and destination in a standard format used in multistate demographic analysis.

Value

`D`	The Biograph object Bdata with diagonal elements removed
`irate`	Option for estimation of rates: (1) Nelson-Aalen, (2) Occurrence-exposure rates, (3) both
`NeAa`	Nelson-Aalen estimator
`predicted`	Cumulative transition rates (Nelson-Aalen estimator) predicted at consecutive ages (using predict function of mvna package)
`astr`	Age-specific transition rate by age, derived from predicted cumulative transition rates

oeCum	Cumulative occurrence-exposure rates
oe	The M-matrix of age-specific transition (occurrence-exposure) rates by origin and destination

Note

The function takes time to execute. It calls Parameters, date_convert, Remove.intrastate, statesequence.ind, Occup, Trans, RateTable, Rates.ac, Biograph.mvna and the mvna and predict functions of the mvna package.

Author(s)

Frans Willekens

Examples

```
data(GLHS)
Bdata.a <- date_b (GLHS,format.out="age",covs=c("marriage","LMentry"))
cumrates <- Cumrates (irate = 3,Bdata=Bdata.a)
```

Date_as_age *Converts a vector of dates (class 'Date') to ages.*

Description

Converts a vector of dates (class 'Date') to ages. Function needs birth dates.

Usage

```
Date_as_age (x,format.in,born)
```

Arguments

x	Vector of dates of class 'Date'
format.in	date format
born	Vector of dates of birth. Birth dates have the same format as the other dates.

Value

age.sec	Age in seconds
age.day	Age in days
age	Age in years, months and days (produced by package lubridate)
age.year	Age in decimal year

Author(s)

Frans Willekens

Examples

```
Date_as_age ("2010-10-30","%Y-%m-%d","1990-4-20")
```

Date_as_cmc *Converts a vector of dates (class 'Date') to Century Month Codes.*

Description

Converts a vector of dates (class 'Date') to months elapsed since 1st january 1900.

Usage

```
Date_as_cmc (x,format.in)
```

Arguments

x	Vector of dates. The Date vector is of class 'Date'
format.in	Format of date

Value

cmc	Dates in cmc
selectday	Day of the month

Author(s)

Frans Willekens

Examples

```
Date_as_cmc("2010-10-30","%Y-%m-%d")
    # cmc is 1330 and day of month is 30
```

Date_as_year *Converts a vector of dates (class 'Date') to decimal year (calendar year and fraction of year).*

Description

Converts a vector of dates (class 'Date') to calendar years and fractions of year.

Usage

```
Date_as_year (x,format.in)
```

Arguments

x	Date as variable of class 'Date'
format.in	Format of the date.

Value

Calendar year and fraction of year

Author(s)

Frans Willekens

Examples

```
Date_as_year (x="1946-03-01")
```

date_b *Converts dates in a Biograph object to dates in another format*

Description

Converts dates in a Biograph object to dates in another format. The function converts an entire Biograph object. It calls the function date_convert, which converts a vector of dates.

Usage

```
date_b(Bdata,selectday,format.out,covs)
```

Arguments

Bdata	Biograph object
selectday	If input date is in Century Month Code (year and month), the argument selectday gives the day of the month. Default value is one, the transition occurs on the 1st of the month. The argument may be omitted.
format.out	Format of output date
covs	Vector of covariates that are dates (that need to be converted too). The dates should be in the same format as the other dates.

Value

Biograph object with dates in new format (format.out)

Author(s)

Frans Willekens

Examples

```
 data(GLHS)
bb<- date_b (Bdata=GLHS,selectday=1,format.out="day-mon-year",covs=c("marriage","LMentry"))
```

---

date_convert *Converts vector of dates from one format to another*

Description

Converts dates at transition to dates in another format or to ages

Usage

```
date_convert(d,format.in,selectday,format.out,born,format.born)
```

Arguments

d	A vector of dates to be converted
format.in	Format of input date
selectday	If input date is in CMC, selectday gives the day of the month. Default is 1
format.out	Format of output date
born	Date of birth for computation of age. Format the date of birth is format.in. May be omitted, unless date at transition need to be converted to age
format.born	Format of date of birth

Value

date	Dates at transition. Object of class 'Date'(number of days since 1970-01-01)
datep	Dates for printing

Author(s)

Frans Willekens

Examples

```
 data(GLHS)
b <- date_convert(d='01/01/2011',format.in='%m/%d/%Y',selectday=1,format.out='day-month-year')
 dd <- GLHS$marriage
 fo <- 'day-mon-year'
 gb <- GLHS$born
bb<- date_convert(d=dd,format.in="CMC",selectday=1,format.out=fo,born=gb,format.born="CMC")
```

GLHS *GLHS data*

Description

data from Blossfeld-Rohwer 2002

Usage

```
data(GLHS)
```

Format

A data frame with 201 observations on the following 24 variables. Dates are given in Century Month Code (CMC).

`ID` Identification number

`born` Month of birth (CMC)

`start` Onset of observation (CMC)

`end` End of observation (CMC)

`sex` Sex. A factor with levels `Males` `Females`

`edu` Years of education.

`marriage` Date of marriage (CMC) (TB in BR data)

`LMentry` Date of entry in labour force (CMC) (TE in BR data)

`cohort` Birth cohort. A factor with levels `1929-31` `1939-41` `1949-51`

`path` State sequence (a character vector)

`Tr1` Date of first transition (CMC)

`Tr2` Date of second transition (CMC)

`Tr3` Date of third transition (CMC)

`Tr4` Date of fourth transition (CMC)

`Tr5` Date of fifth transition (CMC)

`Tr6` Date of sixth transition (CMC)

`Tr7` Date of sevent transition (CMC)

`Tr8` Date of eight transition (CMC)

`Tr9` Date of ninth transition (CMC)

`Tr10` Date of tenth transition (CMC)

`Tr11` Date of eleventh transition (CMC)

`Tr12` Date of twelfth transition (CMC)

Source

For the original data, see http://oldsite.soziologie-blossfeld.de/eha/tda/cf_files/Data/RRDAT.1

References

Blossfeld, H.P. and G. Rohwer (2002) Techniques of event history modeling. New approaches to causal analysis. Lawrence Erlbaum, Mahwah, New Jersey (2nd Edition).

Blossfeld, H.P., K. Golsh and G. Rohwer (2007) Event history analysis with Stata. Erlbaum, Mahwah, New Jersey.

`GLHS.IllnessDeath` *Illness-death model for job transitions*

Description

The function creates from the GLHS data file a Biograph object with the following states (episodes): first job, second or higher-order job and out-of-a-job.

Usage

```
GLHS.IllnessDeath(GLHS)
```

Arguments

`GLHS`	The GLHS data file in Biograph format

Details

The illness-death model is applied to labour-market transitions. In an illness-death model, a person may reach an end-state directly or indirectly by first experiencing an intermediate transition. The illness-death model is generally used in epidemiology and public health. Biograph includes the model to illustrate the mstate package,in particular the msprep function.

Value

The function returns a Biograph object with 3 states: First job, Second or higher-order job and No job.

Author(s)

Frans Willekens

Examples

```
data (GLHS)
ild <- GLHS.IllnessDeath (GLHS)
```

`GLHS.trans` *Transition matrix of illness-death model (tmat)*

Description

The function produces the transition matrix of an illness-death model with three states: Job1, Job2+ and NoJob. The function is called by GLHS.IllnessDeath().

Usage

```
GLHS.trans(names)
```

Arguments

`names`	The names of the three states. If the names are missing, the default names Job1, Job2 and NoJob are given

Value

The transition matrix of possible transitions in the illness-death model. A possible transition is denoted by a number.

Examples

```
data (GLHS)
z <- GLHS.trans()
```

`Lexis.lines` *Draws Lexis diagram with lifelines for selected subjects*

Description

The function draws a Lexis diagram and draws lifelines for selected subjects (using ggplot2 package)

Usage

```
Lexis.lines(Bdata,Dlong, subjectsID, title)
```

Arguments

`Bdata`	Biograph object
`Dlong`	Input data in long format
`subjectsID`	Lifelines are drawn for subjects, identified by ID, included in the vector subjectsID.
`title`	Title of the Lexis diagram.

Details

The function produces a Lexis object and plots the object for selected subjects. Transitions are marked on the lifelines. It requires data in long format. If the long format is not available, the function calls the function Biograph.long to produce the required object.

Value

`subjectsID`	Subjects selected
`p`	The plot of the lifelines

Note

requires ggplot2

Author(s)

Frans Willekens

See Also

Lexislines.episodes

Examples

```
#  Example 1: Employment careers
  data(GLHS)
  z<- Parameters (GLHS)
  GLHS.yr <- date_b(Bdata=GLHS,selectday=1,format.out="year")
  D <- Biograph.long (GLHS.yr)
  tit5 <- "Employment careers for a selection of subjects. GLHS"
  subjects <- c(1,78,120,208)
  z <- Lexis.lines (Bdata=GLHS.yr,Dlong=D$Depisode,subjectsID = subjects,title = tit5)

#  Example 2: Long data format need to be obtained
  z <- Lexislines.episodes (Bdata=GLHS.yr,subjectsID = subjects,title = tit5)

#  Example 3: Living arrangements
  data(NLOG98)
  z<- Parameters (NLOG98)
  NLOG98.yr <- date_b(Bdata=NLOG98,selectday=1,format.out="year")
  D <- Biograph.long (NLOG98.yr)
  tit5 <- "Living arrangements for a selection of subjects, NLOG98"
subjectsID <- c(8,96,980,1056,1496,2883)
  z <- Lexis.lines (NLOG98.yr,D$Depisode,subjectsID = subjectsID,title = tit5)
```

Lexis.points *Plots ages and calendar years at transitions in Lexis diagram*

Description

Plots observations in age-time diagram for each category of categorical covariate (using ggplot2)

Usage

```
Lexis.points(Bdata, transition, title, cov,group,legend.pos,pdf)
```

Arguments

Bdata	Biograph object
transition	The transition selected, e.g. "NJ" in GLHS data
title	Title of Lexis diagram
cov	Covariate. Observations are plotted for each category of this covariate. If observations should be plotted for the total sample population, cov = NA
group	Covariate. Graphs for groups are in separate panels.
legend.pos	position of legend
pdf	If graph should be stored in pdf file: pdf=TRUE, else pdf=FALSE

Value

graph

Note

requires ggplot2

Author(s)

Frans Willekens.

See Also

Lexispoints (requires Epi)

Examples

```
#  Example 1: ages at labour market entry, by sex
  data(GLHS)
  #y <- Parameters(GLHS)
  t <- "Date and age at labour market entry by sex and cohort"
  z <- Lexis.points (GLHS,"NJ",title=t,cov="sex",group="cohort",legend.pos="right",pdf=FALSE)

#  Example 2: ages at first birth among married women, by religious denomination
  data(NLOG98)
  y <- Parameters (NLOG98)
  t <- "Date and age at first birth (married women) by religion and cohort"
  z <- Lexis.points (NLOG98,"MK",title=t,cov="kerk",group="cohort",legend.pos="right",pdf=FALSE)
#  Example 3: ages at leaving home, by birth cohort (any destination)
  t <- "Date and age at leaving parental home by religion and education"
  z <- c(0.7,0.2)
  z <- Lexis.points (NLOG98,"H*",title=t,cov="kerk",group="educ",legend.pos=z,pdf=FALSE)
```

Lexislines.episodes *Draws Lexis diagram with lifelines for selected subjects*

Description

The function draws a Lexis diagram, using the Lexis object defined in the Epi package, and draws lifelines for selected subjects.

Usage

```
Lexislines.episodes(Bdata,Dlong, subjectsID, title)
```

Arguments

Bdata	Biograph object
Dlong	Input data in long format
subjectsID	Lifelines are drawn for subjects, identified by ID, included in the vector subjectsID.
title	Title of the Lexis diagram.

Details

The function produces a Lexis object and plots the object for selected subjects. Transitions are marked on the lifelines. It requires data in long format. If the long format is not available, the function calls the function Biograph.long to produce the required format.

Value

Lcoh11	Lexis object that includes all subjects
Lcohh12	Lexis object that includes selected subjects
sub	vector of IDs of subjects selected for the lifelines in the Lexis diagram
k	Number of subjects selected

Note

requires Epi

Author(s)

Frans Willekens

References

Plummer, M. and B. Carstensen (2011) *Lexis: an R class for epidemiolgoical studies with long-term follow-up.* Journal of Statistical Software, 38(5):1-12

Carstensen, B. (2009) *The Epi package.* Available at http://cran.r-project.org/web/packages/Epi/Epi.pdf

See Also

Lexis.lines

Examples

```
#  Example 1: Employment careers
  data(GLHS)
  z<- Parameters (GLHS)
  D <- Biograph.long (GLHS)
  tit5 <- "Employment careers for a selection of subjects. GLHS"
  subjects <- c(1,78,120,208)
  z <- Lexislines.episodes (Bdata=GLHS,Dlong=D$Depisode,subjectsID = subjects,title = tit5)

#  Example 2: Long data format need to be obtained
  z <- Lexislines.episodes (Bdata=GLHS,subjectsID = subjects,title = tit5)

#  Example 3: Living arrangements
  data(NLOG98)
  z<- Parameters (NLOG98)
  D <- Biograph.long (NLOG98)
  tit5 <- "Living arrangements for a selection of subjects, NLOH98"
  subjectsID <- c(8,96,980,1056,1496,2883)
  z <- Lexislines.episodes (NLOG98,D$Depisode,subjectsID = subjectsID,title = tit5)
```

`LexisOccExp` *Displays occurrences, exposures and transition rates in a Lexis diagram*

Description

The Lexis diagram is an age-time framework. The diagram displays transition data in two time dimensions, e.g. calendar date and age. The Lexis class in the EPI package is an object-based framework for managing transition data on multiple time scales. The function displays occurrences, exposures and transition rates for a selected transltion. The user also determines the age and time interval.

Usage

```
LexisOccExp(Bdata, transition, nyear)
```

Arguments

`Bdata`	Data in Biograph format (Biograph object)
`transition`	The transition selected. A transition is identified by the state of origin and the state of destination.
`nyear`	The age and time interval, generally 5 years.

Details

The presentation of data in the Lexis diagram is particularly helpful to explain the concept of occurrence-exposure rate. The functions plots Lexis diagrams for event counts, exposures and transition rates. In addition, it produces an object with as components the event counts, exposures and rates. It also produces a survival object, which can be used for further survival analysis with the survival package, and a Lexis object, which can be used for further analysis with the Epi package.

Value

`surv`	The survival object. The survival object is used as a response variable in survival models (see the survival package in CRAN)
`Lcoh`	The Lexis object. The Lexis object is a key object in the Epi package.
`nevents`	Event counts: number of transitions by age, origin and destination
`ndur`	Exposure times: duration exposed to the risk of experiencing the selected transition.
`rates`	The transition rates (occurrence-exposure rates). They are obtained by dividing the event counts by the exposure times.

Author(s)

Frans Willekens

References

Carstensen, B. (2007) Age-Period-Cohort models for the Lexis diagram. Statistics in Medicine, 26(15):3018-3045.

Carstensen, B. (2013) The Epi package.

Examples

```
# The following example selects the "JN" transitions from the GLHS data
# and produces Lexis diagrams with age and time intervals of 5 years data (GLHS)
  data (GLHS)
  w <- LexisOccExp (Bdata=GLHS,transition= "JN",nyear=5)
```

Lexispoints *Plot event dates and ages in Lexis diagram*

Description

Plots observations in age-time diagram for each category of categorical covariate (using the Lexis function of the Epi package)

Usage

```
Lexispoints(Bdata, transition, title, cov,legend)
```

Arguments

Bdata	Biograph object
transition	The transition selected, e.g. "NJ" in GLHS data
title	Title of Lexis diagram
cov	Covariate. Observations are plotted for each category of this covariate. If observations should be plotted for the total sample population, cov = NA
legend	legend

Value

Lcoh	Lexis object created to plot the observations

Note

requires Epi

Author(s)

Frans Willekens. The Epi package used to plot the Lexis diagram was developed by Carstensen.

References

Plummer, M. and B. Carstensen (2011) Lexis: an R class for epidemiolgoical studies with long-term follow-up. Journal of Statistical Software, 38(5):1-12

Carstensen, B. (2013) The Epi package.

See Also

Lexislines.episodes

Examples

```
#  Example 1: ages at labour market entry, by sex
  library (Epi)
  data(GLHS)
  #y <- Parameters(GLHS)
  t <- "Calendar time and age at labour market entry"
  z <- Lexispoints (Bdata=GLHS,transition="NJ",title=t,cov="sex",legend="topleft")

#  Example 2: ages at first birth among married women, by religious denomination
  data(NLOG98)
  y <- Parameters (NLOG98)
  t <- "Calendar time and age at first birth (married women)"
  z <- Lexispoints (NLOG98,"MK",t,"kerk")
#  Example 3: ages at leaving home, by birth cohort (any destination)
  t <- "Calendar time and age at leaving parental home"
  z <- Lexispoints (NLOG98,"H*",t,"cohort","topleft")
```

`locpath` *Determines location of state sequence in Biograph object*

Description

Determines location of state sequence (path variable)in Biograph object

Usage

```
locpath(Bdata)
```

Arguments

`Bdata`	Biograph object

Author(s)

Frans Willekens

Examples

```
data(GLHS)
loc <- locpath(GLHS)
```

MSLT *Multistate life table: survival function and exposure function*

Description

The function MSLT.S estimates the multistate survival function from cumulative transition rates. The function MSLT.e uses the multistate life table method to estimate expected state occupation times by state.

Usage

```
MSLT.S(rates)
MSLT.e(SS, radix)
```

Arguments

rates	transition rates. Either the Nelson-Aalen estimators or the occurrence-exposure rates.
SS	The multistate survival function. An object of class 'MSLT.S'
radix	The radix, which is the distribution of the population by state at the initial (reference) age

Details

The multistate life table is produced by two functions: (1) MSLT.S produces the multistate survival function and (2) MSLT.e generates expected sojourn times: total and by state. MSLT.S uses the MatrixExp function of the msm package. MSLT.e applies the multistate life table method that incorporates equations that infer expected sojourn times from transition rates and the associated transition probabilities.

Value

The function MSLT.S returns:

S	The multistate survival function
P	Age-specific transition probabilities

The function MSLT.e returns an object with the following components:

L	For each state, the expected sojourn time by age. It is the number of time units (years, months) a person at the beginning of his or her career (e.g. at birth or at a reference age) may expect to spend in the given state during each age interval.

e0	The total expected sojourn time in each state. If life histories are considered from birth, e0 gives the life expectancy by state. If life histories are truncated at a given age, the life expectancy is the total expected sojourn time in a state between birth and that maximum age.
e.p	Population-based life expectancies by age and state. Population-based measures are independent on the state occupied at the reference age.
e.s	Status-based life expectancies by age and state. Status-based measures depend on the state occupied at the reference age.

Author(s)

Frans Willekens

References

Rogers, A. (1975) Introduction to multiregional mathematical demography. Wiley, New York

Willekens, F. (1987) The marital status life-table. In: J. Bongaarts, T. Burch and K.W. Wachter eds. Family demography: models and applications. Oxford: Clarendon Press. pages 125-149.

Palloni, A. (2001) Increment-decrement life tables. In: S.H. Preston, P. Heuveline and M. Guillot Demography. Measuring and modelling population processes. Blackwell, Oxford, Chapter12, pp. 256-272.

See Also

plot.MSLT.S

Examples

```
# a. The multistate life table based on Nelson-Aalen estimators
# The example obtains Nelson-Aalen estimators of the cumulative transition rates
# using the mavna package. It derives the multistate life survival function
# from the transition rates and the expected sojourn times from the survival function.
# The radix is the number of persons by state at the reference age
# (e.g. at the start of the life history).
    data(GLHS)
    z <- Parameters (GLHS)
    cr <- Cumrates (irate=3,Bdata=GLHS)
    S.e <- MSLT.S(cr$astr[,,,1])  # expected
    radix <- c(10000,0)
    mslt.e <- MSLT.e (S.e,radix)
    S.u <- MSLT.S(cr$astr[,,,2])  # upper
    mslt.u <- MSLT.e (S.u,radix)
    S.l <- MSLT.S(cr$astr[,,,3])  # lower
    mslt.l <- MSLT.e (S.l,radix)

# b. The multistate life table based on occurrence-exposure rates
    cr <- Cumrates (irate=3,Bdata=GLHS)
    S <- MSLT.S(cr$oe)
    radix <- c(10000,0)
    mslt <- MSLT.e (S,radix)
```

NLOG98 *NLOG98 Netherlands Family and Fertility Survey 1998*

Description

Life history data of NLOG98: a subsample of 500 of the 5450 respondents of the NLOG98

Usage

```
data(NLOG98)
```

Format

A data frame with 500 observations on the following 23 variables.

`ID` Personal identification number

`born` Date of birth in Century Month Code (CMC)

`start` Date at entry in observation in CMC

`end` Date at exit from observation in CMC

`kerk` Religious denomination: a factor with 5 levels: `(1) no religion` `(2) Roman Catholic` `(3) Protestant` `(4) other` `(5) missing data`

`educ` Level of education: a factor with 6 levels: `(2) Primary education` `(3) Secondary`

`cohort` Birth cohort: a factor with 2 levels: `(1) <1960` `(2) 1960+`

`path` State sequence during observation period

`Tr1` Date of first transition in CMC

`Tr2` Date of second transition in CMC

`Tr3` Date of third transition in CMC

`Tr4` Date of fourth transition in CMC

`Tr5` Date of fifth transition in CMC

`Tr6` Date of sixth transition in CMC

`Tr7` Date of seventh transition in CMC

`Tr8` Date of eighth transition in CMC

`Tr9` Date of nineth transition in CMC

Details

NLOG98 data

Source

Statistics Netherlands. Life history format: Matsuo and Willekens, 2003

References

Matsuo, H., Willekens F. (2003) Event histories in the Netherlands Fertility and Family Survey 1998: A technical report. PRC Research Report 2003-1, Groningen: Population Research Centre, University of Groningen.

Examples

```
data(NLOG98)
```

`Occup` *State occupancies and sojourn times*

Description

Determines states occupied at each age (birthday) and sojourn time in each state during intervals of one-year age.

Usage

```
Occup(Bdata)
```

Arguments

`Bdata`	Biograph object

Value

`state_occup`	For each age: number of subjects by state and number of subjects censored
`st_age_1`	For each subject: state occupied at each consecutive age. This component may be used as input to the TraMineR package.
`sjt_age_1`	For each subject: sojourn time (years) in each state during age intervals of one year.
`tsjt`	For total sample: number of years spent in each state during age intervals of one year.

Author(s)

Frans Willekens

See Also

Trans is used after Occup

Examples

```
data(GLHS)
param <- Parameters(GLHS)
z <- Occup (GLHS)
```

`OverviewEpisodes` *Overview of episodes*

Description

Computes several indicators of the episodes in the life history data: number of episodes, types of episodes (open or closed) and total and mean durations of episodes.

Usage

```
OverviewEpisodes(Bdata,seq.ind)
```

Arguments

`Bdata`	input data: Biograph object
`seq.ind`	individual state sequences. If seq.ind is missing, the function calls Sequences.ind to obtain the individual state sequences.

Details

Open intervals start before onset of observation (left truncated) or continue after observation ends (right censored). Closed intervals start and end during the observation period.

Value

`n`	Sample size
`ne`	Total number of episodes
`nt`	Total number of transitions
`types`	Number of episodes by type [open. closed]
`sojourn`	Total sojourn time in each type of episode

Author(s)

Frans Willekens

References

Willekens (2011)

Examples

```
data (GLHS)
z<- Parameters (GLHS)
seq.ind <- Sequences.ind (GLHS$path,attr(GLHS,"param")$namstates)
z <- OverviewEpisodes(Bdata=GLHS,seq.ind=seq.ind)
```

`OverviewTransitions` *Overview of transitions*

Description

Produces summary indicators of transitions.

Usage

```
OverviewTransitions(Bdata, seq.ind, agetrans)
```

Arguments

`Bdata`	input data
`seq.ind`	individual state sequences. If seq.ind is missing, the function calls Sequences.ind to obtain the individual state sequences.
`agetrans`	Ages at the different transitions. If agetrans is missing, the function calls AgeTrans.

Value

`Ttrans`	Number of transitions by origin and destination and censored cases by state occupied at time of censoring.
`meanage`	Mean age at transition by origin and destinatin and mean age at censoring.

Note

The function checks for availability of numstates (number of states). If numstates is not available, the function Parameters is called.

Author(s)

Frans Willekens

Examples

```
data (GLHS)
z <- Parameters (GLHS)
seq.ind <- Sequences.ind (GLHS$path,attr(GLHS,"param")$namstates)
agetrans <- AgeTrans (GLHS)
overviewT <- OverviewTransitions (GLHS, seq.ind=seq.ind, agetrans)
```

`Parameters` *Life history parameters from data*

Description

Extracts state space and other characteristics from Biograph object

Usage

```
Parameters(Bdata,newnamstates)
```

Arguments

`Bdata`	Biograph object
`newnamstates`	Biograph detect determines the state space from the data set. The sequence of states detected may not be the most logical sequence. newnamstates allows to change the sequence. Default: sequence of labels detected by Biograph.

Details

Parameters extracts useful information and stores it in values returned by the function.

1. nsample: sample size

2. numstates: number of states in the state space (produced by StateSpace)

3. namstates: labels for the states (determined from the character variable <path>) (produced by StateSpace)

4. absorbstates: vector of absorbing states (determined by StateSpace)

6. iagelow: lowest age in the (sample) population (determined from date at entry in observation and date of birth)

7. iagehigh: highest age in the (sample) population (determined from date at exit from observation and date of birth)

8. namage: labels for the single years of age from the lowest age (iagelow) to the highest age (iagehigh)

9. nage: number of age groups: iagehigh - iagelow + 1

10. maxtrans: maximum number of transitions

11. ntrans: number of transitions

12. trans_possible: transition matrix: feasible transitions

12. tmat: matrix with transition numbers

13. transitions: identifications of transitions: number, origin, destination

14. nntrans: transition counts by origin and destination

15. locpat: column number of <path> variable in Biograph object

16. ncovariates: number of covariates

17. covariates: vector of covariate names
18. format.date: format of date variables (chronological objects) in Biograph object
19. format.born: format of date of birth

The parameters and the format of the dates (format.date) are attached to the Biograph object (Bdata) as attributes.

Value

`sample`	Sample size
`numstates`	number of states in state space
`namstates`	names of states
`absorbstates`	names of absorbing states
`iagelow`	lowest age
`iagehigh`	highest age
`namage`	names of age groups
`nage`	number of age groups
`maxtrans`	maximum number of transitions by individual
`ntrans`	number of transitions
`trans_possible`	Origin-destination matrix of logical elements indicating whether transition is allowed or not. In multistate survival analysis, the matrix is known as 'transition matrix'
`tmat`	Origin-destination matrix showing the line numbers of the transitions. A number is allocated to each possible transition. The matrix tmat is attached as an attribute to the data file.
`transitions`	Data frame which gives for each transition the following information: origin and destination in character value and numeric value
`nntrans`	Number of transitions by origin and destination
`ncovariates`	number of covariates
`covariates`	covariate labels
`format.date`	format of date variables in Biograph object
`format.born`	format of date of birth in Biograph object

Author(s)

Frans Willekens

See Also

`StateSpace`

Examples

```
# Example 1
  data(GLHS)
  z <- Parameters(GLHS)
# Example 2
  data (NLOG98)
  z <- Parameters(NLOG98,newnamstates=c("H","A","C","M","K"))
```

`plot.cumrates` *Plots cumulative transition rates*

Description

Plots cumulative transition rates by origin, destination, and age estimated by `Cumrates`. The required input data are pr

Usage

```
## S3 method for class 'cumrates'
plot(x,ptrans,title,...)
```

Arguments

`x`	An object of class `cumrates`, produced by the `Cumrates` function.
`ptrans`	Selection of transitions to be plotted
`title`	Title
`...`	Further arguments to plot

Details

The function plots cumulative transition rates (cumulative hazard rates). If irate = 1 or irate = 3 it plots the Nelson-Aalen estimator at each birthday and adds lower and upper confidence intervals. If irate = 2 or irate = 3, it plots the cumulative occurrence-exposure rates. If irate = 3, the function plots both the Nelson-Aalen estimator (with confidence intervals) and the cumulative occurrence-exposure rate. The variable irate is an argument of the Cumrates function and included in the object produced by Cumrates.

Author(s)

Frans Willekens

Examples

```
data(GLHS)
Bdata.a <- date_b (GLHS,format.out="age",covs=c("marriage","LMentry"))
cumrates <- Cumrates (irate = 3,Bdata=Bdata.a)
z<- plot (x=cumrates,ptrans=c("NJ","JN"))
```

`plot.MSLT.S` *Plots survival function.*

Description

Plot method for object of class 'MSLT.S' using the ggplot2 package. It plots the multistate survival function by as estimated by `MSLT.S`

Usage

```
## S3 method for class 'MSLT.S'
plot(x, e0, order,colours,title,area,xmin,xmax,...)
```

Arguments

`x`	The multistate survival function. An object of class `MSLT.S` produced by the `MSLT.S` function. It is one of two components of the object produced by the `MSLT.S` function.
`e0`	Life expectancy at reference age (e.g. at birth)
`order`	Desired sequence of states in plot
`colours`	Colours
`title`	Title for multistate survival plot
`area`	Graph type: area (area=TRUE) or bar (area=FALSE)
`xmin`	Minimum value on x-axis
`xmax`	Maximum value on x-axis
`...`	Further arguments to plot

Value

The function plot.MSLT.S returns the multistate survival function (S) and the plot. It returns an object with the following components:

`S`	The multistate survival function (values used in the plot)
`plot`	The plot produced by the ggplot2 package.

Author(s)

Frans Willekens

See Also

MSLT.S

Examples

```
# The multistate life table based on occurrence-exposure rates
    data (GLHS)
    param <- Parameters (GLHS)
    cr <- Cumrates (irate=3,Bdata=GLHS)
    S <- MSLT.S(cr$oe)
    radix <- c(10000,0)
    mslt <- MSLT.e (S,radix)

# Plot the multistate survival function (object of class 'MSLT.S' )
   z<- plot (x=S$S,e0=mslt$e0,title="Multistate survival function",area=TRUE,order=c("N","J"))
```

---

plot.occup.S *Plots state occupancies*

Description

Plots state occupancies. It plots the state occupancies (counts in sample or population under observation) as estimated by Occup. The data are produced by the Occup function.

Usage

```
## S3 method for class 'occup.S'
plot(x,namstates.desired,colours,title,area,xmin,xmax,...)
```

Arguments

x	State occupancies in sample population, by age. An object of class occup.S produced by the Occup function.
namstates.desired	Desired sequence of states in plot. The argument is used to specify an informative ordering of the state occupancies or state probabilities to be stacked.
colours	Colours selected to distinguish the states in the state space.
title	title of plot
area	logical variable.If area is TRUE, area plot is displayed (using geom_area of ggplot2). If area is FALSE, a bar plot is displayed (using geom_bar of ggplot2)
xmin	Minimum age in plot
xmax	Maximum age in plot
...	Further arguments to plot

Details

The function uses the ggplot2 package

Value

occup.S	State occupancies
plot	The figure of state occupancies

Author(s)

Frans Willekens

Examples

```
data(GLHS)
Bdata.a <- date_b (GLHS,format.out="age",covs=c("marriage","LMentry"))
occup <- Occup(Bdata.a)
dd <- occup$state_occup
t <- "States occupancies. GLHS"
cc <- c("red","green","lightgrey")
xx <- c("N","J","Censored")
z<- plot (x=dd,namstates.desired=xx,colours=cc,title=t,area=TRUE,xmin=10,xmax=55)
```

---

pos.char *Position of a given character in a string variable (first match)*

Description

Determines the position(s) of a given character in a string variable. If the character is absent, NA is returned.

Usage

```
pos.char(string, char)
```

Arguments

string	Character string
char	The character to position of which needs to be determined

Details

The function converts the string to a vector of characters. An alternative is to use which(strsplit(string, '')[[1]]=='a'), where a is the character. That code determines all positions of the character in the character string.

Value

The position is returned

Author(s)

Frans Willekens

See Also

printf

Examples

```
string  <- "tests"
pos.char(string,"s")
```

`pos.charstr` *Position of a character string in a string variable*

Description

Determines the position of a character string in a string variable

Usage

```
pos.charstr(string, charstr)
```

Arguments

`string`	The character string
`charstr`	The character variable

Value

First occurrence

Author(s)

Frans Willekens

Rates.ac *Estimates occurrence-exposure rates*

Description

Estimates occurrence-exposure rates. The transition rates are of the age-cohort type.

Usage

```
Rates.ac (Stable)
```

Arguments

Stable	Stable is object produced by RateTable function

Details

The transition rates and the cumulative transition rates are organized as follows: row variable is age, column variable is destination state and layer variable is origin state. The age-cohort transition rates are used to construct multistate life tables.

Value

M	Transition rates by origin, destination and age in standard format [M-format]
Mcum	Cumulative occurrence-transition rates

Author(s)

Frans Willekens

Examples

```
#   Example 1: Transition rates between NOJOB and JOB, based on GLHS
   data(GLHS)
   z<- Parameters (GLHS)
   occup <- Occup(GLHS)
   seq.ind <- Sequences.ind (GLHS$path,attr(GLHS,"param")$namstates)
   trans <- Trans (GLHS)
   ratetable <- RateTable (GLHS,occup=occup,trans=trans)
   rates <- Rates.ac (Stable=ratetable$Stable)

#   Example 2: Rates of transition between living arrangements, based on NLOG98
   data(NLOG98)
   data <- NLOG98[!is.na (NLOG98$kerk)&NLOG98$kerk=="Roman Catholic",]
   z <- Parameters (data)
   seq.ind <- Sequences.ind (data$path,namstatesnew=c("H","C","A","M","K"))
   occup <- Occup (data)
   trans <- Trans (Bdata=data)
   ratetable <- RateTable (NLOG98,occup=occup,trans=trans)
```

```
rates <- Rates.ac (Stable=ratetable$Stable)
```

RateTable *Table for rate calculation*

Description

Ratetable produces a table of transition counts and exposure times for the computation of transition rates (occurrence-exposure rates)

Usage

```
RateTable(Bdata,occup, trans)
```

Arguments

Bdata	Biograph object
occup	Object produced by the Occup function
trans	Object produced by the Trans function

Details

RateTable is one of the key functions of the package. The Stable object it produces contains the data needed for the computation of transition rates by origin, destination and age

Value

Stable	The table of transitions and exposure times
censored_by_age	Number of censored cases by age and state at censoring

Author(s)

Frans Willekens

References

Willekens (2011)

Examples

```
data (GLHS)
z<- Parameters (GLHS)
occup <- Occup(GLHS)
ist <- Sequences.ind (GLHS$path,attr(GLHS,"param")$namstates)
trans <- Trans (GLHS)
w <- RateTable(GLHS,occup, trans)
```

Remove.intrastate *Removes intrastate transitions from Biograph object*

Description

Removes intrastate transitions.

Usage

```
Remove.intrastate(Bdata)
```

Arguments

Bdata	Biograph object

Details

Intrastate transitions are removed and new characteristics of the data set are obtained.

Value

Biograph object with intrastate transitions removed. The "param" attribute is adjusted.

Author(s)

Frans Willekens

Examples

```
data (GLHS)
Bdata2 <- Remove.intrastate (GLHS)
```

rrdat *The Blossfeld-Rohwer subsample of the German Life History Survey (GLHS)*

Description

The data provides information on 600 job episodes.

Usage

```
data(rrdat)
```

Details

ID Identification number of subject

NOJ Serial number of the job episode

TS Starting time of the job episode

TF Ending time of the job episode

SEX Sex (1 male; 2 female)

TI Date of interview (CMC)

TB Date of birth (CMC)

T1 Date of entry into the labour market (CMC)

TM Date of marriage (CMC) [0 if not married]

PRES Prestige score of current job, i.e. of job episode in current record of data file

PRESN Prestige score of the next job (if missing: -1)

EDU Highest educational attainment before entry into labour market

Source

http://oldsite.soziologie-blossfeld.de/eha/tda/cf_files/Data/RRDAT.1

References

Blossfeld, H.P. and G. Rohwer (2002) Techniques of event history modeling. New approaches to causal analysis. Lawrence Erlbaum, Mahwah, New Jersey (2nd Edition).

See Also

GLHS

Examples

```
data(rrdat)
```

`SamplePath` *Obtains samplepath for selected subjects*

Description

Obtains empirical lifepaths for selected subjects

Usage

```
SamplePath(Bdata, subjectsID)
```

Arguments

Bdata	Biograph object
subjectsID	Vector with IDs of selected subjects

Details

Displays lifepaths for selected individuals

Author(s)

Frans Willekens

Examples

```
data (GLHS)
z <- Parameters (GLHS)
   subjectsID <- c(1,6,7,19,136,208)
   samplepaths <- SamplePath (Bdata=GLHS,subjectsID=subjectsID)
```

Sequences *Lists state sequences in the data*

Description

Produces a frequency table of the state sequences recorded in the data. The mean or median ages at transition are computed.

Usage

```
Sequences(Bdata,mean_median)
```

Arguments

Bdata	Biograph object
mean_median	Select mean or median age: "mean" or "median"

Details

The sequences are determined from the path variable in the data. The default age is the median age.

Value

Frequency table of sequences

Author(s)

Frans Willekens

Examples

```
# Example 1: Sequences in GLHS
data(GLHS)
z <- Parameters (GLHS)
seq <- Sequences (Bdata=GLHS,mean_median="median")

# Example 2: Sequences in NLOG98
data(NLOG98)
z <- Parameters (NLOG98)
seq <- Sequences (NLOG98) # default is median age

# Example 3: State sequence in NLOG98, by birth cohort
table(NLOG98$cohort)   # cohort size
seq.1 <- Sequences(NLOG98[NLOG98$cohort=="<1960",],"median")
seq.2 <- Sequences(NLOG98[NLOG98$cohort=="1960+",],"median")
# Note that the sum(seq.2$ncase) equals size of second cohort
```

Sequences.ind *Individual state trajectories*

Description

Shows for each subject in the sample the state sequence (trajectory) from onset to end of observation.

Usage

```
Sequences.ind(path,namstatesnew)
```

Arguments

path	State sequence
namstatesnew	state labels (labels of single characters)

Details

Individual sequences are basis for exploratory sequence analysis

Value

The value returned is a two-way table with for each subject the sequences of states occupied between onset and end of observation

Author(s)

Frans Willekens

Examples

```
data(GLHS)
z <- Parameters (GLHS)
seq.ind <- Sequences.ind (path=GLHS$path,namstatesnew=attr(GLHS,"param")$namstates)
```

`Sequences.ind.0` *Sorts dates in ascending order and generates state sequence (path).*

Description

Sorts dates in ascending order: earlier dates come first. The function also generates state sequences (life path or career path). The function is used in generating a Biograph object from raw data. The state sequence is the path variable of the Biograph object.

Usage

```
Sequences.ind.0 (d,namstates,absorb)
```

Arguments

`d`	Vector of dates
`namstates`	state labels (labels of single characters)
`absorb`	Vector of absorbing states

Value

`namstates`	Labels of the states in the state space.
`d`	Sorted dates
`path`	Character variable of state sequences. This variable is the 'path' variable in the Biograph object.

Author(s)

Frans Willekens

Examples

```
d <- c(300,250,340)  # dates in CMC
f<- Sequences.ind.0 (d,c('A','B','C'))
```

`StateSpace` *Get state space*

Description

Obtains statespace from the data

Usage

```
StateSpace(d,newnamstates)
```

Arguments

`d`	Biograph object (e.g. Bdata) or vector of state sequences (e.g. Bdata$path)
`newnamstates`	The preferred sequence of states.

Details

StateSpace derives statespace from the character vector 'path' in the Biograph object. In that vector, each state is uniquely identified by a single character (ASCII). The function finds the different characters. The sequence of states is determined by characters popping up first, second, etc. You may impose a preferred sequence by specifying the argument newnamstates. If no preferred sequence exists, newnamstates is missing from the argument list. The results are stored in two variables: numstates (number of states) and namstates (names of the states). The function also returns an object identifying the absorbing state (if any). A state that is entered but not left is an absorbing state.

Value

`namstates`	Names of the states in the state space
`absorbstates`	List of absorbing states. It is NULL if there is no absorbing state.

Author(s)

Frans Willekens

Examples

```
data(GLHS)
z<- StateSpace (GLHS)
data (NLOG98)
z<- StateSpace (NLOG98,c("H","A","C","M","K"))
z <- StateSpace (NLOG98$path)
```

`state_age` *Determines for given individuals state occupied at given ages*

Description

Determines state occupied at given exact ages (birthdays). The function is used in the Occup function

Usage

```
state_age (Bdata,age,ID)
```

Arguments

`Bdata`	Biograph object
`age`	Vector of ages for which state occupancy should be determined
`ID`	Vector of IDs of individuals for whom state occupancy should be determined

Value

`nam`	Names of states that can be occupied. The list includes '-' to indicate that at that age observation did not start yet and '+' to indicate that observation ended.
`state`	State occupied at given age by each individual in the study.
`state.n`	State occupied at given age: all individuals (table).

Author(s)

Frans Willekens and Sabine Zinn

See Also

Occup

Examples

```
data(GLHS)
param <- Parameters(GLHS)
agetrans <- AgeTrans(Bdata=GLHS)
z <- state_age (Bdata=GLHS,age=c(20,30),ID=c(15,208))
```

state_time *Determines individual state occupation times*

Description

Determines for given individuals state occupation times at all ages. The function is used in the Occup function

Usage

```
state_time (Bdata,ID)
```

Arguments

Bdata	Biograph object
ID	Vector of IDs of individuals for whom state occupancy should be determined

Value

state	Individual state occupancies at each birthday
state.n	State occupancies by age
sjt_age_1	For each subject selected: occupation time in each state during age intervals of one year (time in years).
tsjt	For total of selected individuals: number of years spent in each state during age intervals of one year (all ages) (time in person-years).

Author(s)

Frans Willekens and Sabine Zinn

See Also

Occup

Examples

```
data(GLHS)
param <- Parameters(GLHS)
zz <- state_time (Bdata=GLHS,ID=c(15,208))
```

string.blank.omit *Removes blanks in character string, including the leading and trailing white space.*

Description

Removes blanks in character string and returns string without blanks.

Usage

```
string.blank.omit(string)
```

Arguments

string	Character string

Value

The function returns the string with blanks removed.

Author(s)

Frans Willekens

Examples

```
z <- string.blank.omit ("This is a test   ")
y <- string.blank.omit (" A B   C  ")
```

stringf *Converts character string to vector*

Description

Converts character string to vector

Usage

```
stringf(string)
```

Arguments

string	character string

Details

Converts string

Value

The function returns a vector with elements the characters of a character string.

Author(s)

Frans Willekens

Examples

```
z <- stringf("test")
```

Trans *Transitions by age*

Description

Produces tables of transitions: (1) by origin and destination and (2) by origin, destination and age. It also computes the mean ages at transition.

Usage

```
Trans(Bdata)
```

Arguments

Bdata	Biograph object

Value

Ttrans	Total number of transitions by origin and destination.
meanage	Mean age of transitions by origin and destination
trans	Number of transitions by origin, destination and age
states_during_interval	Number of states occupied during an age interval of one year by an average subject.

Author(s)

Frans Willekens

Examples

```
data (GLHS)
y<- Parameters(GLHS)
z <- Trans (GLHS)
```

`TransitionAB` *Occurrences of selected transition by age*

Description

Determines for a given transition the number of transitions by age

Usage

```
TransitionAB(Bdata, transition,keep)
```

Arguments

`Bdata`	Biograph object
`transition`	Selected transition
`keep`	Logical variable indicating whether observations (subjects) with the selected transition missing should be kept (TRUE) or removed (FALSE). Default: keep=FALSE

Value

`case`	The transition
`n`	Number of subjects in the sample experiencing the selected transition (transition count).
`id`	Identification number of subjects that experienced the transition
`pos`	Position of the selected transition in the state sequence (path variable Bdata$path of the Biograph object). If transition = "*N" and N is the first state, then that state is skipped. If a subject experienced several of the selected transition, the first transition is considered.
`date`	For each subject experiencing the transition, date of transition
`age`	For each subject experiencing the transition, age at transition
`year`	For each subject experiencing the transition, year of transition (year is real variable, including fraction of year)
`cohort`	For each subject experiencing the transition, birth cohort

Author(s)

Frans Willekens

Examples

```
# Example 1: Transition NJ in GLHS data set
  data (GLHS)
  z <- TransitionAB (GLHS,"NJ",keep=TRUE)

# Example 2: Transition HM in NLOG98 data set
```

```
  data (NLOG98)
  z <- TransitionAB (NLOG98,"HM")

# Example 3: Transition 'Leaving parental home", irrespective of destination state
  data (NLOG98)
  z <- TransitionAB (NLOG98,"H*")

# Example 4: First marriage, irrespective of origin state
  data (NLOG98)
  z <- TransitionAB (NLOG98,"*M")
```

`transitions` *Generate flow table*

Description

Generates table of transitions by origin and destination.

Usage

```
transitions(Bdata,newnamstates)
```

Arguments

`Bdata`	Biograph object
`newnamstates`	New names of the states

Value

`nsample`	Sample size
`namstates`	Names of the states
`ntrans`	Number of possible interstate transitions
`nntrans`	Transition count by origin and destination
`trans_possible`	Logical variable indicating which transitions are feasible
`transitions`	List of feasible transitions: line number of transition, origin state and destination state
`tmat`	Line numbers of the feasible transitions

Author(s)

Frans Willekens

Examples

```
data (GLHS)
z <- transitions (GLHS)
```

YearTrans *Converts dates at transition to years at transition*

Description

Converts dates at transition to decimal years at transition (years and fraction of year)

Usage

```
YearTrans(Bdata)
```

Arguments

Bdata	Biograph object

Value

For each subject, years at transition

Note

For each subject under observation, the years of the following events are given: birth, entry into observation, end of observation, each transition.

Author(s)

Frans Willekens

See Also

CMC_as_year and AgeTrans

Examples

```
data(GLHS)
yeartrans <- YearTrans(GLHS)
```

year_as_age *Converts decinal years to age.*

Description

Converts a vector of decimal years to ages. The function requires birth dates.

Usage

```
year_as_age (x,born,format.born)
```

Arguments

x	Vector of decimal years.
born	Vector of dates of birth. Class is 'Date'
format.born	format of dates of birth.

Value

age

Author(s)

Frans Willekens

Examples

```
year_as_age (2010.578,"1990-10-30","%Y-%m-%d")
    # result: 19.7506
```

year_as_cmc *Converts decinal years to cmc.*

Description

Converts a vector of decimal years to CMC.

Usage

```
year_as_cmc (x)
```

Arguments

x	Vector of decimal years.

Value

cmc

Author(s)

Frans Willekens

Examples

```
year_as_cmc (2010.578)
```

`year_as_Date` *Converts a vector of dates in decimal years (calendar years and fractions of year) to object of class 'Date'.*

Description

Converts vector of calendar years and fractions of year to object of class 'Date'.

Usage

```
year_as_Date (x,format.out)
```

Arguments

`x`	Calendar year and fraction of year, e.g. 2012.448
`format.out`	Format of date

Value

date as object of class Date: days since 1-1-1970 and printed as date in Gregorian calendar, e.g. "1946-03-01"

Author(s)

Frans Willekens

Examples

```
# d=1946.167, which is March 1,1946
year_as_Date (x=1946.167,format.out='%d-%m-%Y')
```

Index