Package 'Families'

April 13, 2024

Type Package

Title Kinship Ties in (Virtual) Multi-Generation Populations

Version 2.0.2

Depends R (>= 4.3.0)

Imports msm,reshape,ggplot2,rlang

Suggests knitr, rmarkdown,lubridate,xml2,plyr,VirtualPop

BuildResaveData best

LazyData true

Date 2024-04-13

Maintainer Frans Willekens <willekens@nidi.nl>

Description Tools to study lineages, grandparenthood, loss of close relatives, kinship networks and other topics in multi-generation populations.

License GPL-2

NeedsCompilation no

Encoding UTF-8

BugReports https://github.com/willekens/Families/issues

RoxygenNote 7.2.3

Author Frans Willekens [aut, cre] (<https://orcid.org/0000-0001-6125-0212>)

Repository CRAN

Date/Publication 2024-04-13 07:50:10 UTC

R topics documented:

`Families-package` *Kinship Ties in Virtual Populations*

Description

Tools to study kinship networks, grandparenthood, loss of close relatives and double burden (presence of children and oldest old parents) in virtual population produced by 'VirtualPop'.

Author(s)

Frans Willekens <Willekens@nidi.nl>

`Age` *Age of ego or kin at given calendar date*

Description

Ages are computed from calendar dates.

Usage

```
Age(idref, refT)
```

Arguments

`idref`	vector of IDs of reference population, usually idego
`refT`	Calendar date (decimal date) provided or computed

Value

`Age`	Age of kin (or ego). If ego is not yet born or dead at the reference date, the missing data symbol NA is returned.

Examples

```
# Load data
data(dLH,package="Families")

# Age of ego on January 1, 2100. Ego is individual with ID equal to 1 in dLH
idego <- 1
age <- Age(idref=idego,refT=2100)

# Age of ego at death of mother
idego <- dLH$ID[dLH$gen==2]
age <- Age(idref=idego,refT=Dd(IDmother(idego)))

# Age of siblings at ego's 20th birthday
idego <- dLH$ID[dLH$gen==2]
# Get IDs of siblings and convert the list object into a dataframe
idsib <- IDsib(idego)
names(idsib$id) <- idego
dfsib <- Kin_long(idsib$id)
colnames(dfsib) <- c("idego","idkin")
# Get ages of siblings at 20th birthday of egos
dfsib$age <- Age(idref=dfsib$idkin,refT=Db(dfsib$idego)+20)
```

Alive *Living status of individual at given age or calendar date*

Description

Determines living status of individual at given age or calendar date. The living status is requested for ego or kin. If the number provided is less than 200, the number is interpreted as age.

Usage

```
Alive(idref, refA)
```

Arguments

idref	vector of IDs.
refA	calendar date (decimal date) or age at which survival status of individuals should be determined. Dates can differ between individuals, but number of dates must be same as number of individuals.

Value

alive_kin	Living status of ego (if id is a vector) or kin (if id is a dataframe)

Examples

```
# Load the data
data(dLH,package="Families")
# Select members of generation 1
idego <- dLH$ID[dLH$gen==1]
# Are egos alive at age 18?
alive <- Alive(idref=idego,refA=18)
# Number alive and deceased
tab <-  table (alive)

## Is ego alive at 18th birthday of oldest child?
date <- sapply(idego,function(x) min(Db(IDch(x))+18))
alive <- Alive(idego,refA=date)
```

Db *Date(s) of birth in decimal format*

Description

Retrieves date(s) of birth from the database

Usage

```
Db(idego, d = NULL)
```

Arguments

idego	vector of IDs of egos
d	Name of database. If d is missing, dLH is used.

Value

Dates of birth (decimal format)

Examples

```
# Load the data
data(dLH,package = "Families")

# Date of birth of first individual in database
d <- dLH
Db(idego=1,d=d)
```

Dd *Date(s) of death in decimal format*

Description

Retrieves date(s) of death from the database

Usage

```
Dd(idego, d = NULL)
```

Arguments

idego	vector of IDs of egos
d	Name of database. If d is missing, dLH is used.

Value

Date of death (decimal format)

Examples

```
# Load the data
data(dLH,package = "Families")

# Date of death of first individual in database
Dd(idego=1,d=dLH)
```

dLH *Individual fertility histories of members of a virtual population. Sample dataset.*

Description

Fertility histories based on period data and in the presence of mortality. The histories are simulated from age-specific death rates and conditional fertility rates of USA 2021. The virtual population consists of 5 generations (with selected info on members of the sixth generation). The VirtualPop package is used to generate the population.

Usage

```
data(dLH,package="Families")
```

Format

A data frame with data of about 7,000 individuals (2000 in initial cohort).

ID Identification number

gen Generation

cohort Birth cohort

sex Sex. A factor with levels Males and Females

bdated Date of birth (decimal date)

ddated Date of death (decimal date)

x_D Age at death (decimal number)

IDmother ID of mother

IDfather ID of father

jch Child's line number in the household

IDpartner ID of partner

udated Date of union formation

nch Number of children ever born to woman

Source

The data for the simulation are period mortality rates by age and sex and period fertility rates by age and birth order for the United States 2021. The data are downloaded from the Human Mortality Database (HMD) and the Human Fertility Database (HFD).

IDau *IDs of aunts and uncles of ego*

Description

Retrieves the IDs of ego's aunts and uncles by blood. Partners (in-laws) are not included. They are obtained using the IDPartner() function.

Usage

```
IDau(idego, d = NULL)
```

Arguments

idego	Identification number(s) of ego(s).
d	Name of database. If missing, dLH is used, if it exists in the global environment (i.e. R workspace). Otherwise the dataset dLH distributed with the Families package are used.

Value

IDs of aunts and uncles by blood (without their partner, i.e. in-laws).

Examples

```
# Load the data
data(dLH,package="Families")
set.seed <- 45
idego <- sample(dLH$ID[dLH$gen==3],1)
idau <- IDau (idego)

idego <- dLH$ID[dLH$gen==3]
idau <- IDau (idego,d=dLH)
```

IDch *IDs of children of ego*

Description

Retrieves IDs of children of ego(s).

Usage

```
IDch(idego, d = NULL, keep_ego = FALSE)
```

Arguments

idego	ID of ego(s). If the vector idego includes missing elements (NA), they are removed.
d	Name of database. If d is missing, the dataset dLH in the global environment (R workspace) is used. If no dLH in the global environment, the database dLH distributed with the Families package is used.
keep_ego	Logical variable. If TRUE, a dataframe of parent-child and father-child dyads is produced. It includes, for each ego (parent), ego's ID and the IDs of ego's children.

Value

Two cases:

- keep_ego=FALSE: IDch() returns the IDs of children. If ego has no children or IDs of children are not included in database, the missing data symbol NA is returned. The vector idego may include the IDs of egos who form a couple. In that case, the IDs of their children are included only once to prevent double-counting.

- keep_ego=TRUE: IDch() returns a dataframe of parent-child dyads. If idego includes the IDs of egos who form a couple, then the IDs of their children are included twice, in the mother-child dyads and in the father-child dyads. To select the father-child dyads, select the male egos. The dataframe of parent-child dyads include childless females and males. The dyad has the ID of the female(male) and NA instead of the ID of the child. The object returned has the following columns:
  - ID of ego (parent of child)
  - ID of child

Examples

```
# Load the data
data(dLH,package=("Families"))

IDch(idego=1)
set.seed(43)
id <- sample (dLH$ID[dLH$gen==1],10)
id2 <- IDch(idego=sort(id),keep_ego=TRUE)
id3 <- IDch(id2$idch,keep_ego=TRUE)
```

IDfather *ID of father of ego*

Description

Retrieves the ID of the father of ego or fathers of vector of egos. ID of the father is the ID of the partner of the mother. If the ID of the father listed in the dataset d differs from the ID of the mother's partner, a warning is given.

Usage

```
IDfather(idego, d = NULL)
```

Arguments

idego	ID of ego(s)
d	Name of database. If missing the dataset dLH distributed with the Families package is used.

Value

ID of father or (if keep_ego=TRUE, dataframe with ego-father dyals: ID of ego and ID of father). Returns NA if ID of father is not included in the database

Examples

```
# Load the data
data(dLH,package = "Families")

set.seed(31)
idf <- IDfather (idego=sample (dLH$ID[dLH$gen>=2],10))
```

IDgch *IDs of grandchildren of ego*

Description

Retrieves ID of grandchildren of ego or vector of egos

Usage

```
IDgch(idego, d = NULL, keep_ego = FALSE)
```

Arguments

idego	ID of ego(s)
d	Name of database. If d is missing, the dataset dLH in the global environment (R workspace) is used. If no dLH in the global environment, the database dLH distributed with the Families package is used.
keep_ego	If keep_ego=TRUE, parent-child-grandchild triads are shown.

Value

Two cases:

- keep_ego=FALSE: The function IDgch() returns the IDs of grandchildren. If ego has no grandchildren or IDs of grandchildren are not included in database, the missing data symbol NA is returned.
- keep_ego=TRUE: IDgch() returns a data frame of child-parent-grandparent triads. A triad consists of:
  - idego ID of grandparent (ego)
  - idch ID of child
  - idgch ID of grandchild
  - gp lineage: maternal grandfather, paternal grandmother, etc.
  - idMOM ID of mother of grandchild
  - idDAD ID of father of grandchild

Examples

```
# Load data
data(dLH,package="Families")
IDch(IDch(dLH$ID[dLH$sex=="Female" & dLH$gen==1]))[1]

set.seed(51)
id <- sample (dLH$ID[dLH$gen==1],10)
id2 <- IDgch(idego=sort(id),keep_ego=TRUE)
```

IDmother *ID of mother of ego*

Description

Retrieves the ID of mother of ego or mothers of vector of egos

Usage

```
IDmother(idego, d = NULL)
```

Arguments

idego	ID
d	Name of database. If d is missing, the dataset dLH in the global environment (R workspace) is used. If no dLH in the global environment, the database dLH distributed with the Families package is used.

Value

ID of mother. Returns NA if ID of mother is not included in the database

Examples

```
# load the data
data(dLH,package = "Families")
IDmother (sample(dLH$ID[dLH$gen==2],1),d=dLH)
```

IDpartner *ID of partner of ego*

Description

Retrieves ID of partners of vector of egos #'

Usage

```
IDpartner(idego, d = NULL)
```

Arguments

idego	vector of IDs of egos.
d	Name of database. If d is missing, the dataset dLH in the global environment (R workspace) is used. If no dLH in the global environment, the database dLH distributed with the Families package is used.

Value

Vector of IDs of partners

Examples

```
# Load the data
data(dLH,package = "Families")
# Get ID of partner
IDpartner(idego=1,d=dLH)
IDpartner(idego=c(4,9,NA,30),d=dLH)
```

IDsib *IDs of siblings of ego*

Description

Retrieves IDs of siblings of ego

Usage

```
IDsib(idego, d = NULL)
```

Arguments

idego	ID of ego
d	Name of database. If d is missing, the dataset dLH in the global environment (R workspace) is used. If no dLH in the global environment, the database dLH distributed with the Families package is used.

Value

A list vector of two elements.

- dfsib: dataframe of IDs of ego-sibling dyads.
- id: vector of IDs of siblings.

Returns NA if ID of sibling cannot be computed because the ID of mother is not included in the database.

Examples

```
# Load data
data(dLH,package="Families")
# IDs of siblings of single member of generation 3
set.seed(34)
idego <- sample(dLH$ID[dLH$gen==3],1)
idsib <- IDsib (idego)

# For each member of generation 2, IDs of siblings
idego <- dLH$ID[dLH$gen==2]
idsib <- IDsib(idego)
```

Kin_long *Converts a list vector of kin into a dataframe of kin*

Description

Converts a list vector of kin into a dataframe of kin. The dataframe has a "long format", i.e. it has one row for each ego-kin dyad.

Usage

```
Kin_long(idkin)
```

Arguments

idkin	List vector of kin

Value

Dataframe of kin (long format)

Examples

```
# Load data
data(dLH,package="Families")
# IDs of ego and their grandchildren
idego <- dLH$ID[dLH$gen==1]
# IDs of grandchildren of ego
idgch <- lapply(idego,function(x) IDch(IDch(x)))
names(idgch) <- idego
# Dataframe with ID of grandmother and grandchild
dfgch <- Kin_long(idkin=idgch)[,1:2]
```

Nkin *Number of kin of given type*

Description

Computes, for each ego, number of kin of given type (parents, grandparents, aunts, uncles, siblings, etc.)

Usage

```
Nkin(idkin)
```

Arguments

idkin	dataframe of IDs of ego and IDs of kin of given type

Value

Number of kin of given type

Examples

```
# Load the data
data(dLH,package="Families")
idego <- dLH$ID[dLH$gen==3]
## Number of siblings
idsib <- IDsib(idego)
names(idsib$id) <- idego
nsib <- Nkin(idkin=idsib$d)
```

PlotAges *Plots age distribution(s) of kin*

Description

Plots one or more age distributions of kin (e.g. males, females; mothers, grandmothers)

Usage

```
PlotAges(d, xmin = NULL, xmax = NULL, ymax = NULL, legendPos = NULL)
```

Arguments

d	Dataframe with at least the following columns: • ID • Age • Case. Case is the population category for which age distribution should be drawn (e.g. male, female).
xmin	Minimum age to be displayed on x-axis
xmax	Maximum age to be displayed on x-axis
ymax	Maximum value on y-axis (minimum is 0)
legendPos	Position of legend (position is indicated according to ggplot2 rule)

Value

p	syntax to plot age distribution(s)

Examples

```
# Load data
data(dLH,package="Families")
# Age of mother at birth of a child
idego=dLH$ID[dLH$gen==1 & dLH$sex=="Female"]
idch <- IDch(idego,d=dLH)
agem <- dLH$bdated[idch]  - dLH$bdated[IDmother(idch,d=dLH)]
dm <- data.frame(idego=IDmother(idch),Age=agem)
dm$Case <- "Motherhood"

# Age at grandmotherhood
idgch <- IDch(IDch(idego,d=dLH),d=dLH)
agegm <- dLH$bdated[idgch] - dLH$bdated[IDmother(IDmother(idgch,d=dLH),d=dLH)]
dgm <- data.frame(idego=IDmother(IDmother(idgch,d=dLH),d=dLH),Age=agegm)
dgm$Case <- "Grandmotherhood"
d <- rbind (dm,dgm)
d <- d[!is.na(d$Age),]
binwidth <- (max(d$Age,na.rm=TRUE)-min(d$Age,na.rm=TRUE))/60
cas <- unique(d$Case)
```

```
d$Case <- factor(d$Case,levels=cas,labels=cas,ordered=TRUE)
library(ggplot2)
p <- PlotAges(d)
```

`Tests` *Checks the input data*

Description

Tests existence of data file and identifies anomalies. If the data file d is null, the function checks whether dLH exists in the global environment. If it does, d=dLH. If not, the message "Object dLH is not provided and does not exist" is displayed and the processing stops. If some values in idego vector are missing (are NA) a message is displayed and the missing elements are removed.

Usage

```
Tests(idego, d = NULL)
```

Arguments

`idego`	ID
`d`	Name of database. If d is missing, the dataset dLH in the global environment (R workspace) is used. If no dLH in the global environment, the database dLH distributed with the Families package is used.

Details

The function Tests() is called in functions starting with ID, e.g. IDmother(). If the idego vector has values outside the acceptable range or missing values (NA), these outvalues are omitted and Tests gives a warning. The warning is not exported to the function calling Tests(). It is recommended to call Tests() before any other function call.

Value

idego and d

Index