Families and Kinship Networks in Virtual Populations. The 'Families' package.

Frans Willekens

2022-07-06

Contents

1 Introduction

Families are social networks of related individuals belonging to different generations. A social network may be approached from a *holistic perspective*, which concentrates on network characteristics, or an *egocentric* perspective, which concentrates on network ties of a reference person or focal individual, called *ego*. Ego may be a member of the oldest generation, the youngest generation, or an intermediate generation. That leads to different perspectives on population. If ego is a member of the oldest generation, ego's perspective is that of a parent, a grandparent and a great-grandparent. If, on the other hand, ego is a member of the youngest generation, the perspective is that of a child, grandchild and great-grandchild. Members of an intermediate generation are both parents and children. Some of these members may go through a stage of life with old parents and young children and experience the double-burden of caring for parents and children. To study family ties from different perspectives, simulation is often used. Simulation produces a virtual population that mimics a real population. The multi-generation virtual population offers unique opportunities to study aspects of family demography that did not receive much attention until recently: (a) the study of the population from a child's perspective, (b) the demography of grandparenthood (perspective of the elderly), and (c) the double burden experienced by the sandwich generation.

The three subjects listed are receiving a growing attention in the demographic literature. The dominant method of analysis is microsimulation and SOCSIM is the software package of choice. SOCSIM creates individual life histories and genealogies from age-specific mortality and fertility rates (Wachter et al. 1997). 'VirtualPop', a simulation package in R (**Willekens2022package?**), uses a similar method to produce individual life histories and genealogies in multigeneration populations. There is an important difference, however: simulation proceeds in continuous time. Microsimulation in continuous time has two important advantages (Willekens 2009). First, no two events can occur in the same interval, which resolves the issue of event sequence that requires the user to determine which event comes first and which second. In SOCSIM, the simulation proceeds month by month and events scheduled in a month are executed in random order (Wachter et al. 1997, p. 93)[1]. Second, the duration between events can be computed exactly, not approximately.

Microsimulation is often used to study subjects that cannot be studied easily otherwise. Wolf (1994), Arpino et al. (2018), Margolis (2016), Margolis and Verdery (2019) and others adopt the perspective of the elderly, and raise issues such as the presence of children and grandchildren, the number of grandchildren, the timing of grandparenthood, the duration of grandparenthood, and the age of grandchildren at a particular moment in a grandparent's life. A second subject is population structure and demographic change from the perspective of children. A child interprets demography as the presence of parents, grandparents, siblings and peers. The perspective of children is rarely adopted in demography (Mills and Rahal 2021, p. 20). Lam and Marteleto (2008) and Verdery (2015) adopt the perspective of a child to interpret demographic change. Using survey data, Bumpass and Lu (2000) study the children's family contexts in the United States. The third subject is the double burden of child care and elderly care. The double burden was recently studied by Alburez-Gutierrez et al. (2021). The authors used mortality and fertility rates of 198 countries and territories included in the 2019 Revision of the UNWPP and the period 1950-2100 to estimate the time spent caring for as child (and grandchild) and an elderly parent. The output is a complete kinship network for the 1970–2040 birth cohorts from which it is possible to determine the time each simulated individual spent sandwiched as a parent and grandparent. The authors conclude that "Demographic microsimulation allowed us to overcome the lack of international and comparable data on past, present, and future kinship structures."

The purpose of the vignette is to shown how to navigate the multi-generation virtual population database to extract demographic information of direct interest to aspects of family demography listed above. The desired information is extracted using queries, formulated as R functions. The code uses subsetting, which is a feature of vectorised operations. Most of the functions in R are vectorised, which means that a function operates on all elements of a vector simultaneously. Vectorized operations prevent the loops that makes R slow[2].

This document consists of eight sections in addition to the introduction. The next section is a very brief introduction to 'VirtualPop'. Section three presents six basic queries. They are sufficient to retrieve families ties and produce kinship networks. Section four combines the basic functions in queries to obtain information on the kin of ego: siblings, cousins and aunts. The fifth section zooms in on grandparenthood. It illustrates how the basic queries may be used to answer questions about grandparenthood, such as age at onset and duration. Section six concentrates on the double burden. It addresses questions such as age at onset and termination, and the proportion of people who experience a double burden. The last section presents the fertility table, which is a multistate life table of fertility histories. The application of the basic functions,

[1] In the Modgen microsimulation package developed by Statistics Canada, simulations proceeds in continuous time (for a brief description, see Belanger and Sabourin (2017, p. 14ff)).

[2] The simulation of the fertility careers of 50,000 women, their daughters, granddaughters and great-granddaughters takes about 2 minutes on a MacBook.

alone or in combination, to extract desired information on families and kinships in the virtual population may be labeled *kinship calculus*.

2 The 'VirtualPop' package

The 'VirtualPop' package generates a multi-generation virtual population. You select the number of generations and the size of the first generation (generation 1). The sizes of the other generations depend on the fertility rates. For each individual in the virtual population, the lifespan and the fertility history from death are generated by sampling age-at-death distributions implicit in the death rates by age (single years of age) and sex included in the Human Mortality Database (HMD) and the time-to-event (waiting time) distributions implicit in the fertility rates by age and birth order. Rates may be downloaded from the Human Fertility Database (HFD). 'VirtualPop' uses *conditional fertility rates* (in HFD parlance). They are more widely known as occurrence-exposure rate. For details, see the tutorial and the other vignettes included in the 'VirtualPop' package.

Note that the life histories of members of the virtual population produced by 'VirtualPop' are not projections or forecasts. They are the life histories and genealogies that would result if the individuals in successive generations experience the same mortality and fertility. The demographic indicators computed from the histories convey information on the actually observed mortality and fertility rates, unless the rates are programmed to change in time. A virtual population is analogous to a stable population. In demography, stable population theory is used to assess the long-term consequences of sets of age-specific fertility and mortality rates, and rates of transition between living states. In the long run, life history characteristics and population characteristics coincide (ergodicity theorem). The same theorem applies to a multi-generation virtual population that results from a set of age-specific fertility and mortality rates, and rates of transition between living states. In the long run, i.e. after a large number of generations, life history characteristics and population characteristics coincide.

The fertility and mortality rates of the United States in the calendar year 2019 and the virtual population generated by the 'VirtualPop' from these rates are included in the data folder of the 'Families' package. To list the datasets in the package, type $data(package =' Families')$. To attach the package and to load the data, the following code is used:

```
# Attach the package to the search path
library (Families)
# Load the data in your workspace (environment:R_GlobalEnv)
data(dataLH_F,package = 'Families')
data(rates,package = 'Families')

dLH <-  dataLH_F
# Number of generations in the dataset
ngen <- max(unique (dLH$gen))
```

Note that the virtual population in *dLH* differs from the data file distributed with "VirtualPop$ due to randomness. The virtual population is produced from the same rates, but with a different random seed. The virtual population consists of 2965 individuals belonging to 4 generations. The following table shows the number of individuals by generation and sex:

```
addmargins(table (Generation=dLH$gen,Sex=dLH$sex))
#>           Sex
#> Generation Male Female  Sum
#>          1  535    465 1000
#>          2  416    368  784
#>          3  324    343  667
#>          4    266    248  514
#>          Sum 1541   1424 2965
```

3 Basic functions to navigate the virtual population database

The key to retrieve information on a reference person or multiple reference persons is the individual identification number (ID). The ID of a reference person is denoted by $IDego$ or $idego$. Since R treats a scalar as a vector, a request for information on a single reference person or on a group of reference persons is treated similarly. $IDego$ (and $idego$) is a vector of reference persons. In case of a single reference person, the vector consists of a single element. To retrieve data on an individual's social network (kinship network), the 'Families' package includes four basic functions to navigate the virtual population database. They retrieve the IDs of mother, father, partner and children. A combination of basic functions retrieves the IDs of siblings, aunts, uncles and cousins. In addition to these four functions, two basic functions return dates. In multi-generation populations, working with dates is more convenient than working with ages. The six functions are:

- $IDmother(idego)$ retrieves the ID of ego's mother.
- $IDfather(idego)$ the ID of ego's father
- $IDpartner(idego)$ retrieves the ID of ego's partner.
- $IDch(idego)$ retrieves the IDs of ego's children
- $Db(idego)$ retrieves the decimal date of birth
- $Dd(idego)$ retrieves the decimal date of death

Ego's age at an event is the date of birth minus the date of the event. For instance, the age at death is the date of birth minus the date of death: $Db(idego) - Dd(idego)$.

To retrieve information on kin other than members of the nuclear family, the basic functions are combined. For instance, the function call $IDmother(IDmother(idego))$ gives the ID of ego's grandmother and $IDmother(IDmother(IDmother(idego)))$ retrieves the ID of ego's great-grandmother. If an individual for which information is requested is not included in the database, the missing value indicator NA is returned. The query $IDfather(IDmother(idego)))$ returns the ID of ego's maternal grandfather and $IDfather(IDfather(idego)))$ returns the ID of the paternal grandfather. The function call $IDch(IDch(idego))$ returns the IDs of ego's grandchildren, and $IDch(IDch(IDch(idego)))$ the IDs of the great-grandchildren.

The names of the basic functions may not be unique to 'Families'. Other packages included in the Comprehensive R Archive Network (CRAN) may have function by the same name. CRAN has about 20 thousand contributed packages. To ensure that $IDmother$ and the other basic functions attached during the computations are from 'Families' and not from another package in the archive, the double colon operator is used. The second line of the code chunk that follows ensures that $IDmother$ of the package 'Families' is attached and not a function by the same name in another package.

In the remainder of this section, information is retrieved on mothers, grandmothers, children and grandchildren.

3.1 Mothers and grandmothers

By way of illustration, let's retrieve the data on three individuals, their mother and maternal grandmother. Before doing that, the functions of the 'Families' package, which are in the package environment, are copied to the workspace, which is the global environment.

```
# Create local copies of the functions (in workspace)
IDmother <- Families::IDmother
IDfather <- Families::IDfather
IDpartner <- Families::IDpartner
IDch <- Families::IDch
Db <- Families::Db
Dd <- Families::Dd
```

The individuals are selected at random from members of the third generation. The retrieval of information on mother and grandmother requires the retrieval of (a) the IDs of mother and grandmother, and (b) the individual records of these persons. The code is:

```
base::set.seed(30)
idego <- sample (dLH$ID[dLH$gen==3],3)
z <- dLH[c(idego,IDmother(idego),IDmother(IDmother(idego))),]
rownames(z) <- NULL
z[,1:9]
#>     ID gen    sex   bdated   ddated      x_D IDpartner IDmother IDfather
#> 1 2114   3   Male 2083.456 2150.064 66.60759      2444     1409     1248
#> 2 2037   3 Female 2072.402 2156.915 84.51251      1971     1346     1675
#> 3 2090   3   Male 2084.189 2149.381 65.19175      1791     1389     1017
#> 4 1409   2 Female 2052.485 2138.830 86.34457      1248      512      536
#> 5 1346   2 Female 2041.926 2121.007 79.08010      1675      433      946
#> 6 1389   2 Female 2046.990 2133.761 86.77143      1017      487       87
#> 7  512   1 Female 2019.307 2090.162 70.85469       536       NA       NA
#> 8  433   1 Female 2019.066 2106.830 87.76340       946       NA       NA
#> 9  487   1 Female 2019.613 2111.424 91.81184        87       NA       NA
```

The three individuals selected have IDs 2114, 2037, 2090. The first three records contain data on the reference persons. Records 4-6 have data on the mothers of the reference persons and records 7-9 have data on the grandmothers. For each individual, the table includes ID of the individual, generation, sex, date of birth, date of death and age of death. In addition, it includes the ID of the partner, and the IDs of the mother and father (if their generation is included in the virtual population). Individual records also include the birth order of the individual and data on her or his children: the number of children (nch), the IDs of the children and the ages of the mother at childbearing. In this R R chunk, the information is limited to the first 3 children:

```
z[,c(10:14,21:23)]
#>   jch nch id.1 id.2 id.3    age.1    age.2    age.3
#> 1   1   0   NA   NA   NA       NA       NA       NA
#> 2   3   2 2643 2644   NA 25.15381 25.68533       NA
#> 3   5   0   NA   NA   NA       NA       NA       NA
#> 4   2   4 2114 2115 2116 30.97098 31.44769 34.29536
#> 5   1   3 2035 2036 2037 23.89045 25.05770 30.47579
#> 6   1   6 2086 2087 2088 27.01064 28.33435 29.80034
#> 7  NA   2 1408 1409   NA 32.19153 33.17808       NA
#> 8  NA   2 1346 1347   NA 22.86008 25.85480       NA
#> 9  NA   3 1389 1390 1391 27.37700 36.22489 40.23379
```

The second individual has 2 children. The ID of the children are 2462, 2463 and the age of the mother at birth of the child is 29.43, 33.92 years.

To list the IDs of egos, their mothers and grandmothers, the function call is:

```
IDmother(IDmother(idego,keep_ego=TRUE))
#>   IDgm IDmother IDego
#> 1  512     1409  2114
#> 2  433     1346  2037
#> 3  487     1389  2090
```

with IDgm the ID of the grandmother of ego. The second argument of the function instructs the function to keep the IDs of ego and ego's mother. For an explanation of the code, see the description of the function $IDmother()$ (by typing ?IDmother into the console).

3.2 Children and grandchildren

Individual with ID 2 has 2 children and 2 grandchildren. The IDs of the grandchildren are:

```
# Children
IDch(id=2)
#> [1] 1663
# Grandchildren
IDch(IDch(id=2))
#> numeric(0)
```

Consider all women in generation 1 (4883 women) and let *idego* denote the vector of IDs of the women. The IDs of women with children are:

```
# Select all females in generation 1
idego <- dLH$ID[dLH$gen==1 & dLH$sex=="Female"]
# IDs of children
idch <- IDch(idego)
# IDs of mother with children
idm <- unique(IDmother(idch))
```

with idm the IDs of women with children. 3751 women have children and 1132 remain childless. The IDs of women without children are retrieved by the typing $idego[idego\%in\%idm == FALSE]$. The proportion remaining childless is $1 - length(idm)/length(idego)$, which gives 22.37 percent. The number of women in generation 1 by number of children is

```
addmargins (table(dLH$nch[dLH$gen==1 & dLH$sex=="Female"]))
#>
#>   0   1   2   3   4   5   6 Sum
#> 104 114 131  75  27   9   5 465
```

The IDs of the oldest and youngest child in a family are relatively easily determined. The oldest child is the child with the lowest date of birth. For each woman in the generation 1, the ID of the oldest child is obtained as follows:

```
idego <- dLH$ID[dLH$gen==1 & dLH$sex=="Female"]
# ID of children
idch <- IDch(idego,dLH)
# Date of birth of children
dbch <- dLH$bdated[idch]
# Create data frame
```

```r
zz <-data.frame (ID=idch,dbch=dbch)
# Select, for each ego, child with lowest date of birth
ch_oldest=aggregate(zz,list(dLH$IDmother[idch]),function(x) min(x))
colnames(ch_oldest) <- c("idego","ID of oldest child","date of birth of oldest child")
```

The object *ch_oldest* is a data frame with three columns. The first has the ID of the mother, the second the ID of the oldest child, and the third the date of birth of the oldest child. The first lines of *ch_oldest* are

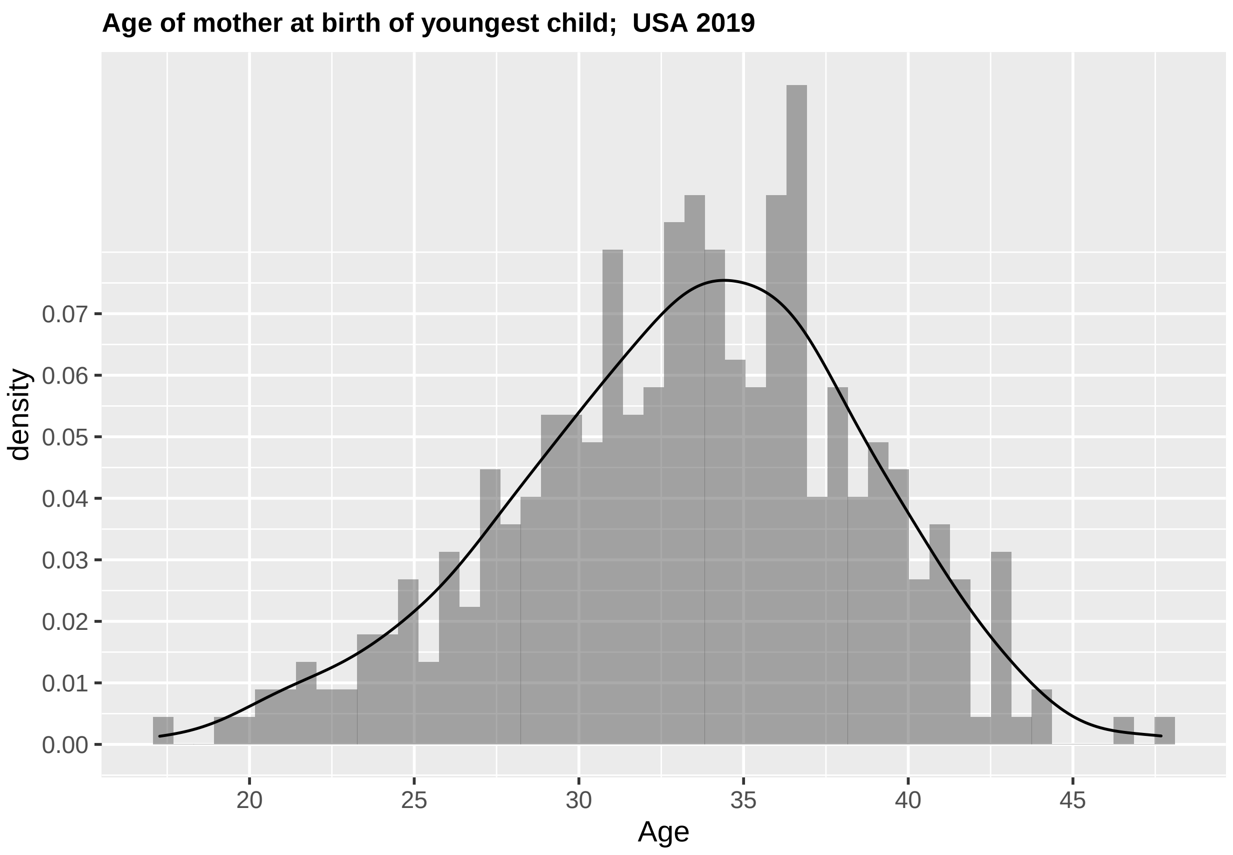
```
#>   idego ID of oldest child date of birth of oldest child
#> 1     1               1001                      2051.490
#> 2     3               1003                      2042.449
#> 3     6               1004                      2038.721
#> 4     7               1006                      2058.484
#> 5     9               1008                      2043.317
#> 6    10               1010                      2037.625
```

For each woman with children, the ID of the youngest child is

```r
# Select, for each ego, child with highest date of birth
#    zz has ID of all children and dates of birth of children.
#    They are used to select youngest child ever born (by mother)
ch_youngest=aggregate(zz,list(dLH$IDmother[idch]),function(x) max(x))
# Date of birth of mother
ch_youngest$db_mother <- Db(ch_youngest[,1])
# Age of mother at birth youngest child
ch_youngest$agem_chLast <- ch_youngest[,3] - ch_youngest$db_mother
colnames(ch_youngest) <- c("idego","ID_youngest_child","db_youngest_child","db_idego","agemLast")
```

ch_youngest is a data frame with five columns: the ID of the mother, the ID of the youngest child, the date of birth of the youngest child, the date of birth of the mother and the age of the mother at birth of the youngest child. The density distribution of the ages of mothers at birth of youngest child is:

```r
  xmin <- 10
  xmax <- 50
  library(ggplot2)
  p <- ggplot(ch_youngest, aes(agemLast)) +
               geom_histogram(aes(y=..density..), alpha=0.5, position="identity",bins=50)+
               geom_density(alpha=0.2) +
            scale_x_continuous(breaks=seq(xmin,xmax,by=5)) +
            scale_y_continuous (breaks=seq(0,0.07,by=0.01)) +
            labs(x="Age")
  title <- paste ("Age of mother at birth of youngest child; ",attr(dLH,"country"),attr(dLH,"year") )
  p <- p + ggtitle(title) +
  theme(plot.title = element_text(size = 10, face = "bold"))
  p
```



The mean age is 33.31 years and the standard deviation is 5.25 years. The median age at birth of the youngest child is $r round(median(ch_youngest$agemLast),2)‘ years.

3.3 Child’s perspective

The age of a newborn’s mother is the difference between the date of birth of the child and the date of birth of the mother. The age of a mother at a given reference age of a child is the age of the mother at birth of the child plus the reference age, provided the mother and the child are both alive at that age. Suppose we are interested in the age distribution of mothers of 10-year old children. A 10-year old is the reference person (ego). Consider females of generation 1. The IDs of mothers alive at ego’s 10 birthday is:

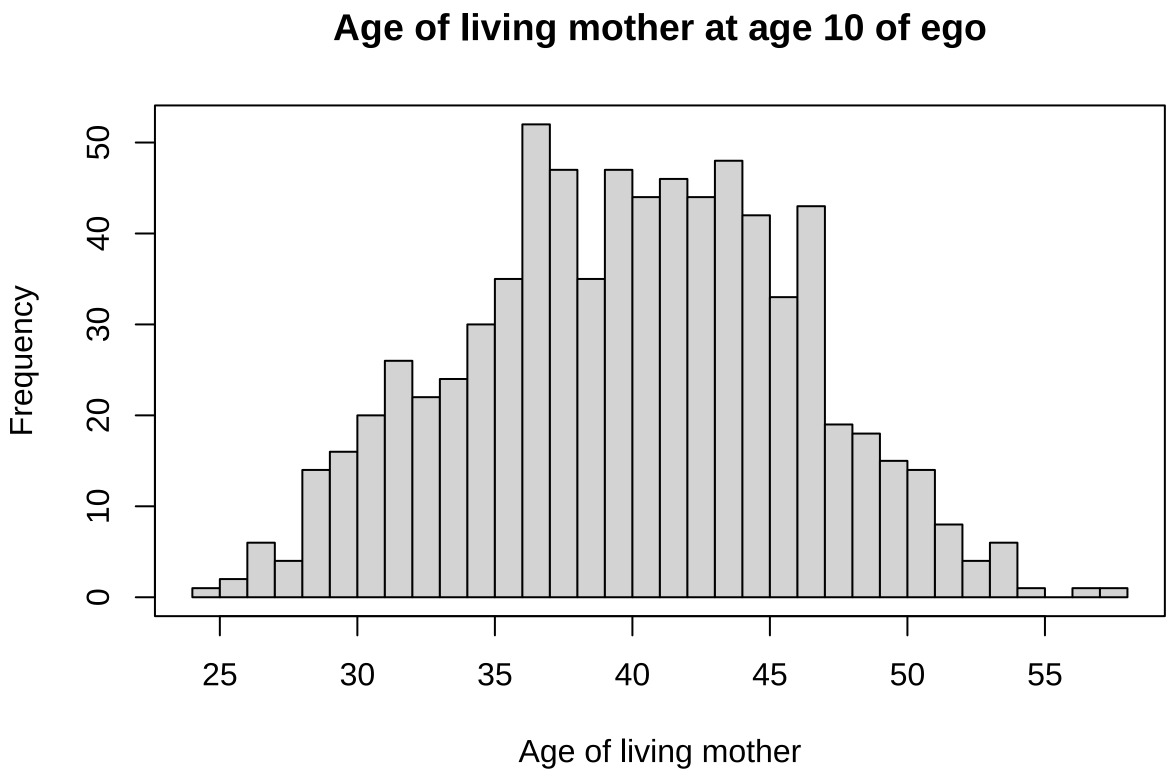
```r
Status_refageEgo <- function(refage_ego)
{ # IDs of egos (children)
  idego <- IDch(dLH$ID[dLH$gen==1 & dLH$sex=="Female"])
  # Is ego alive at reference age?
  alive_ego <- Dd(idego) >= Db(idego) + refage_ego
  # Number of children (ego) alive (=TRUE) and dead (=FALSE) at reference age
  t1 <- table (alive_ego)
  # IDs of egos alive at reference age
  idegoRefage <- idego[alive_ego]
  # Is mother alive at ego's reference age?
  alive_m <- Dd(IDmother(idego)) >=  Db(idego) + refage_ego
  # Number of mothers alive (=TRUE) or dead (=FALSE) at reference ages of egos
  t2 <- table (alive_m)
  aa <- list (idego=idego,
              alive_ego=alive_ego,
              alive_m=alive_m,
              LivingStatus_ch_refage=t1,
              LivingStatus_m_refage=t2)
  return(aa)
}
refage_ego <- 10
out <- Status_refageEgo (refage_ego)
```

The proportion of egos with a living mother at their 10th birthday is

```
round (length(out$alive_m[out$alive_m]) / length(out$alive_ego),2)
#> [1] 0.99
```

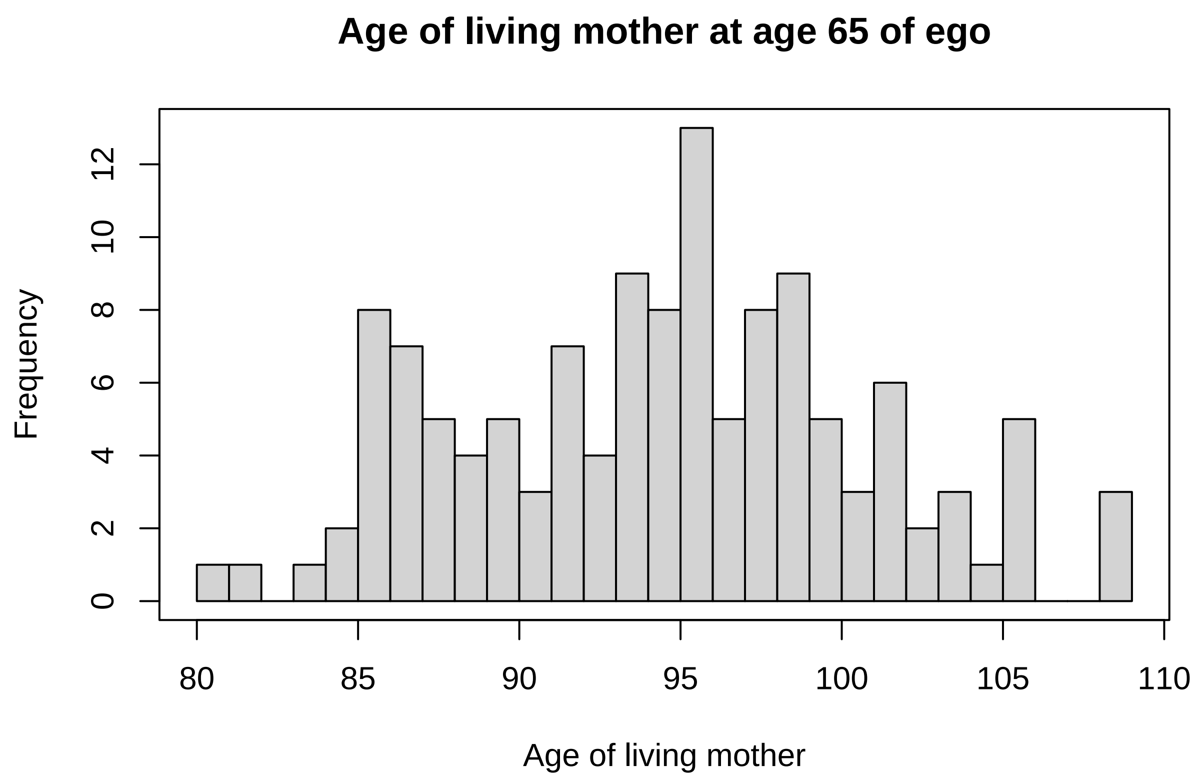
The age distribution of mothers at reference age of children is:

```
# Age of living mothers at refage of ego
idegoRefage <-  out$idego[out$alive_ego]
age_m <- (Db(idegoRefage) + refage_ego - Db(IDmother(idegoRefage)))[out$alive_m]
mean(age_m)
#> [1] NA
sd(age_m)
#> [1] NA
hist(age_m,main=paste ("Age of living mother at age ",refage_ego," of ego",sep=""),breaks=40,xlab="Age o
box()
```



The age distribution of living mothers at age 65 of children is:

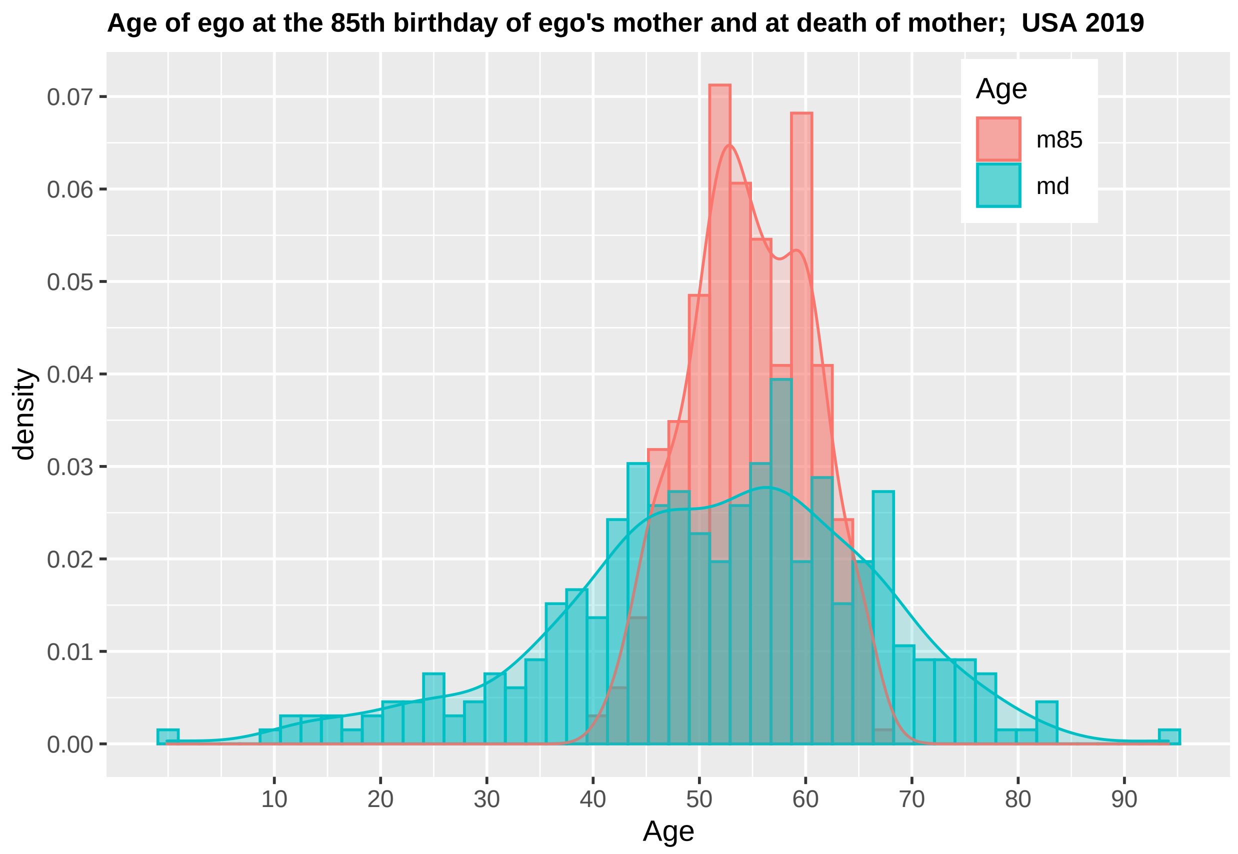
```
refage_ego <- 65
out <- Status_refageEgo (refage_ego)
# Age of living mothers at refage of ego
idegoRefage <-  out$idego[out$alive_ego]
age_m <- (Db(idegoRefage) + refage_ego - Db(IDmother(idegoRefage)))[out$alive_m]
mean(age_m)
#> [1] NA
sd(age_m)
#> [1] NA
hist(age_m,main=paste ("Age of living mother at age ",refage_ego," of ego",sep=""),breaks=40,xlab="Age o
box()
```



The age of a child at a given reference age of the mother is computed in a similar way. The age is the date of birth of the mother plus the reference age minus the date of birth of the mother, provided mother and child are alive. The age of ego at the 85th birthday of ego's mother and the age of ego at death of ego's mother (provided ego is alive at these ages) are:

```
idego <- dLH$ID[dLH$gen==3 & dLH$sex=="Female"]
age_m85 <- dLH$bdated[IDmother(idego)] + 85 - dLH$bdated[idego]
age_md <- dLH$ddated[IDmother(idego)] - dLH$bdated[idego]
d <- data.frame (idego,m85=age_m85,md=age_md)
```

```r
library (ggplot2)
Plot_ages <- function (d)
{ # Age of ego at age 85 of mother and at death of mother
  dd <- reshape::melt.data.frame(d,id.vars="idego",measure.vars=c("m85","md"))
  colnames(dd)[2] <- "Age"
  xmin <- 10
  xmax <- 90
  p <- ggplot(dd, aes(x=value,color=Age,fill=Age)) +
              geom_histogram(aes(y=..density..), alpha=0.5, position="identity",bins=50)+
              geom_density(alpha=0.2) +
           scale_x_continuous(breaks=seq(xmin,xmax,by=10)) +
           scale_y_continuous (breaks=seq(0,0.07,by=0.01)) +
           xlab("Age")
  # Add median
  p <- p + theme(legend.position=c(0.76,0.99),legend.justification=c(0,1))
  title <- paste ("Age of ego at the 85th birthday of ego's mother and at death of mother; ",attr(dLH,"
  p <- p + ggtitle(title) +
  theme(plot.title = element_text(size = 10, face = "bold"))
 p
}
Plot_ages(d)
```



The number of years with a mother aged 85 and over is the mother's remaining number of years at age 85, provided the mother reaches 85. It is:

```
# Age at death
z <-  dLH$ddated[IDmother(idego,dLH)] - dLH$bdated[IDmother(idego,dLH)]
alive <- z >=85
duration <-  dLH$ddated[IDmother(idego,dLH)][alive] - (dLH$bdated[IDmother(idego,dLH)][alive] + 85)
```

The mean duration is 8.26 years and the standard deviation is 5.42 years.

4 Combinations of basic functions

4.1 Siblings

4.1.1 IDs of siblings

A combination of basic functions retrieves the IDs of sisters, brothers, aunts, uncles and cousins. Sisters and brothers of ego have the same mother and cousins have the same grandmother. Aunts and uncles are the siblings of ego's mother and father. The IDs of ego's siblings are retrieved in two steps. First, the IDs of the children of ego's mother are retrieved. Next, the ID of ego is removed.

```
# select one individual of generation 3 who is third child of the family as ego
idego <- sample(dLH$ID[dLH$gen==3 & dLH$jch==3],1)
# IDs of the children of ego's mother
IDch(IDmother(idego))
#> [1] 2106 2107 2108
# ID of ego's siblings: children of ego's mother, excluding ego
IDch(IDmother(idego))[IDch(IDmother(idego))%in%idego==FALSE]
#> [1] 2106 2107
```

Let's considers three unrelated reference persons. The following function call yields the children of the mothers of the reference persons:

```
idego <- c(2723,2192,2291)
IDch(IDmother(idego))
#> [1] 2192 2291 2292 2293 2294 2722 2723
```

The siblings of the three unrelated reference persons are:

```
IDch(IDmother(idego))[IDch(IDmother(idego))%in%idego==FALSE]
#> [1] 2292 2293 2294 2722
```

If two reference persons have the same mother, the approach does not work. Suppose the sister of ID 27023 is included. The siblings produced by the above method the shared siblings only:

```
idego <- c(2723,2724,2192,2291)
IDch(IDmother(idego))[IDch(IDmother(idego))%in%idego==FALSE]
#> [1] 2292 2293 2294 2722 2725 2726
```

In general, if the reference persons include two members of the same family, then the code gives the shared siblings of these two family members. It does not give the siblings for each family member included in the list of reference persons. The reason is that ALL reference persons are excluded from the list of siblings. To obtain a correct result, a list object should be used. The following code produces a list object of four elements, one for each reference person. An element contains the IDs of siblings of one of the reference persons.

```
f <- function(idego)
      y <- IDch(IDmother(idego))[IDch(IDmother(idego))%in%idego==FALSE]
kk <- list()
for (i in 1:length(idego))
{ z <- f(idego[i])
  kk[[i]] <- z
}
names(kk) <- paste ("Siblings of reference person ",idego[1:length(idego)],sep="")
kk
#> $`Siblings of reference person 2723`
#> [1] 2722
#>
#> $`Siblings of reference person 2724`
#> [1] 2725 2726
#>
#> $`Siblings of reference person 2192`
#> numeric(0)
#>
#> $`Siblings of reference person 2291`
#> [1] 2292 2293 2294
```

Consider all members of the second generation in the virtual population. To retrieve their siblings, the code is

```
idego <- dLH$ID[dLH$gen==2]
f <- function(idego)
      y <- IDch(IDmother(idego))[IDch(IDmother(idego))%in%idego==FALSE]
sib <- list()
for (ego in 1:length(idego))
{ z <- f(idego[ego])
  sib[[ego]] <- z
}
```

The first member of the second generation has ID *idego*[1] and the siblings are *sib*[[1]]. The siblings of individual with ID 10001 are:

```
sib[[which(idego==1001)]]
#> [1] 1002
```

In the virtual population, 14 percent of children in the second generation have no siblings, 37 percent have 1 brother or sister, 25 percent have 2 siblings and about one fourth (23.8 percent) have 3 or more siblings. The code to compute the figures is:

```
z <- sapply(sib,function(x) length(x),simplify=TRUE)
percentage <- round (100 * table(z)/sum(table(z)),2)
percentage
#> z
#>     0     1     2     3     4     5
#> 14.54 33.42 28.70 13.78  5.74  3.83
mean <- mean(z,na.rm=TRUE)
sd <- sd(z,na.rm=TRUE)
```

The mean number of siblings is 1.74 and the standard deviation is 1.25.

4.1.2 Age distribution of siblings

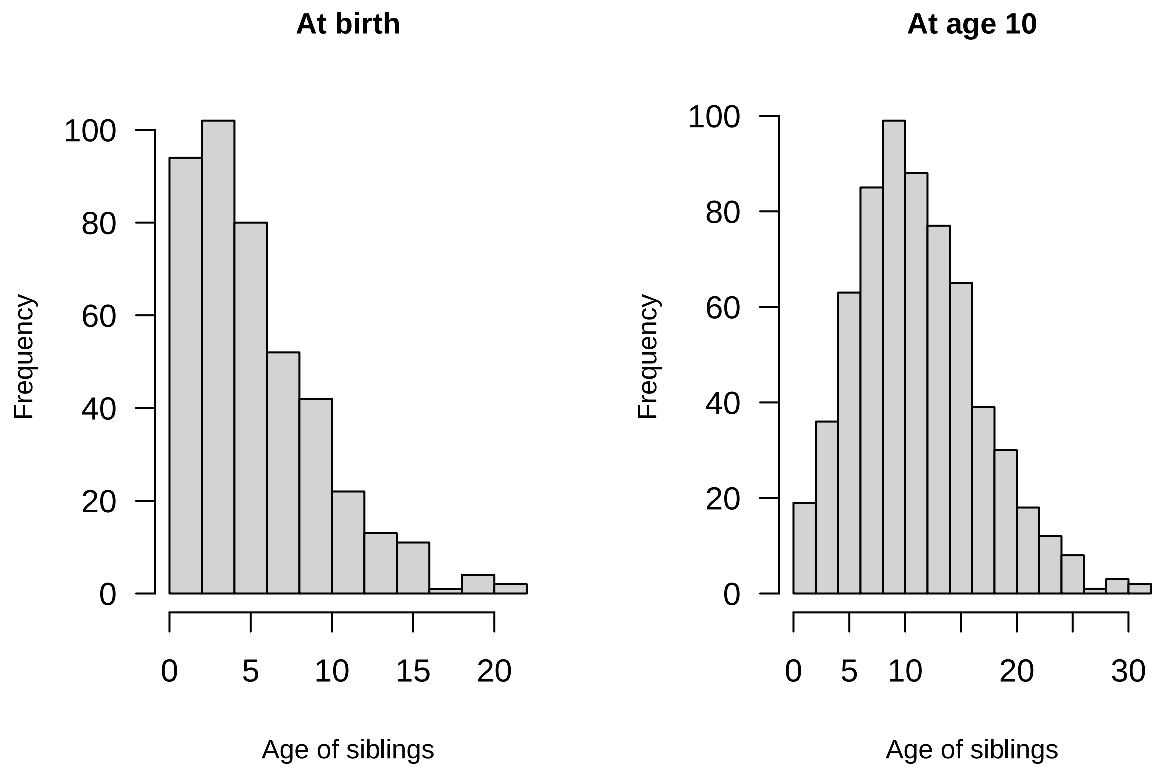
The age distribution of siblings is determined at a given reference age of egos. Consider members of the second generation at ages 0 and 10. The number of individuals is 784.

The code is:

```r
# Ages of siblings at age 10 of ego
m <- as.list(idego)
# Merge two lists: list of egos and list of siblings of egos
zz <- base::Map(c,m,sib)
AgeSib <- function (zz,ageRef)
   { kk <- sapply (zz,function(x)
      { # ageReference <- 30
        # In absence of siblings, skip
        if (length(x)==1) mage <- NA else
        {
        # Date of birth of ego
        db_ego <- Db(x[1])
        # Dates of birth of siblings
        db_sib <- Db(x[2:length(x)])
        # Ages of siblings at reference age of ego
        age_sib <- db_ego+ageRef - db_sib
        # Omit negative values (siblings born after refAgeth birthday of ego)
        age_sib[age_sib<0] <- NA
        # Mean age of siblings at age ageReference of ego
        mage <- mean(age_sib,na.rm=TRUE)
        }
      })
    return(kk)
   }
# Age distribution of siblings at age 10 of ego
mage_sib10 <- AgeSib(zz,ageRef=10)
# Age distribution of siblings at birth of ego
mage_sib0 <- AgeSib(zz,ageRef=0)
```

The age distribution of siblings at birth of ego and the age distribution at age 10 are:

```r
# par(mfrow=c(2,1))
# layout(matrix(1:2,nrow=1),widths=lcm(c(7,8)), heights=lcm(c(9,9)))
layout(matrix(1:2,nrow=1))
hist(mage_sib0,las=1,main="At birth",xlab="Age of siblings",cex.main=0.9,cex.lab=0.8)
hist(mage_sib10,las=1,main="At age 10",xlab="Age of siblings",cex.main=0.9,cex.lab=0.8)
```



The mean age of siblings at age 10 of ego is 11.07 years and the standard deviation is 5.53 years. Note that children without siblings and siblings born after age 10 of ego are not included in the computations. At birth the mean age of siblings is 5.27 years and the standard deviation is 4.04 years. The age difference between ego and siblings is highest at birth and declines during the first years of life to reach 0 around age 20.

4.1.3 Child's perspective

In the second generation, 25 percent of women remain childless. Of women with children, 32 percent have one child, 40 percent two children and 28 percent three or more children. From a woman's perspective, one out of four have one child. From a mother's perspective, one out of three has 1 child.

```
# Select women in generation 2
id <- dLH$ID[dLH$gen==2 & dLH$sex=="Female"]
# Proporiton of women by number of children ever born
z <- table (dLH$nch[idego]) / sum(table (dLH$nch[idego]))
# Proportion of mothers by number of children ever born
zz <- round (100 * table (dLH$nch[idego])[2:6]/sum(table (dLH$nch[idego])[2:6]),2)
```

A child's perspective on family composition differs significantly from the perspective of parents. From a child's perspective, only children represent only 14 percent of all children (see above).

4.2 Cousins

To obtain the IDs of cousins of ego, we first retrieve the IDs of the grandchildren of ego's grandmother and then delete the children of ego's mother (ego and ego's siblings). Consider individual with ID 27023. The person's maternal cousins are:

```
idego <- 2289
idgm <- IDmother(IDmother(idego))
idgch <- IDch(IDch(idgm))
idcousins <- idgch[idgch%in%IDch(IDmother(idego))==FALSE]
idcousins
#> numeric(0)
```

with idgm the ID of ego's grandmother and ch the IDs of the grandchildren of idgm. The subsetting deletes the children of ego's mother. The number of maternal cousins is *length*(*idcousins*).

Consider a random sample of three members of the third generation. The IDs of their maternal cousins are:

```
idego <- sample(dLH$ID[dLH$gen==3],3)
idgm <- IDmother(IDmother(idego))
idcousins <- IDch(IDch(idgm))[IDch(idgm)%in%IDmother(idego)==FALSE]
```

If the selected individuals include individuals with the same mother, then the method works too because all siblings are removed.

To get each reference person's cousins, use a list object:

```
idgm <- IDmother(IDmother(idego))
f <- function(id,idgm)
            { grandch <- IDch(IDch(idgm))
              cous <- grandch[IDmother(grandch)%in%IDmother(id)==FALSE]
            }
idcousins <- list()
for (i in 1:length(idego))
{ z <- f(idego[i],idgm[i])
  idcousins[[i]] <- z
}
```

cousins gives for each selected ego the IDs of the maternal cousins.

The maternal cousins of every member of the third generation (6592 memmbers) are (the computations take time):

```
# IDs of members of the third generation
idego <- dLH$ID[dLH$gen==3]
# IDs of their grandmothers
idgm <- IDmother(IDmother(idego))
# IDs of their cousins
idcousins <- list()
for (i in 1:length(idego))
{ z <- f(idego[i],idgm[i])
  idcousins[[i]] <- z
}
```

An element *idcousins*[[*i*]] of the list object *idcousins* gives the IDs of the cousins of ego i.

The frequency distribution of maternal cousins of members of the third generation is

```
z <- sapply(idcousins,function(x) length(x),simplify=TRUE)
percentage <- round (100 * table(z)/sum(table(z)),2)
percentage
#> z
#>     0     1     2     3     4     5     6     7     8     9    10    12    13
#> 25.64 11.69 15.44 10.64  9.15  6.90  7.50  6.60  3.30  1.35  1.50  0.15  0.15
```

One of four children has no maternal cousins, while 16.8 percent have 2 cousins.

The frequency distribution of paternal cousins is:

```
idego <- dLH$ID[dLH$gen==3]
idgm <- IDmother(IDfather(idego))
idcousins <- list()
for (i in 1:length(idego))
{ z <- f(idego[i],idgm[i])
  idcousins[[i]] <- z
}
z <- sapply(idcousins,function(x) length(x),simplify=TRUE)
percentage <- round (100 * table(z)/sum(table(z)),2)
percentage
#> z
#>     0     1     2     3     4     5     6     7     8     9    10    11    12
#> 27.29  9.00 16.49  7.95 13.79  7.05  3.90  5.40  3.15  1.20  1.80  0.30  1.50
#>    13    14
#>  0.75  0.45
```

The frequency distribution is similar to that of maternal cousins, as expected because of the significance of random factors in the simulation.

4.3 Aunts

Maternal aunts are sisters and sisters-in-law of ego's mother. The IDs are obtained in two steps. First, the IDs of daughters and daughters-in-law of ego's grandmother are determined. Second, the ID of the mother of the reference persons are deleted:

```
# ID of grandmother
idgm <- IDmother(IDmother(idego,dLH),dLH)
# Function of compute the IDs of grandmother's daughters, their partners, and the IDs of aunts
f <- function(id,idgm,dLH)
            { iddaught <- c(IDch(idgm)[dLH$sex[IDch(idgm)]=="Female"],
              IDpartner(IDch(idgm)[dLH$sex[IDch(idgm)]=="Male"]))
              aunts <- iddaught[iddaught%in%IDmother(id)==FALSE]
            }

aunts <- list()
for (i in 1:length(idego))
{ z <- f(idego[i],idgm[i],dLH)
  # IDs of aunts
  aunts[[i]] <- z
}
```

Consider as egos the members of the third generation. The frequency distribution of maternal aunts is

```
z <- sapply(aunts,function(x) length(x),simplify=TRUE)
percentage <- round (100 * table(z)/sum(table(z)),2)
percentage
#> z
#>     0     1     2     3     4     5
#> 15.29 29.09 30.88 14.99  5.85  3.90
```

Fourteen percent have no aunt, 37 percent have one aunt and 25 percent have two aunts.

5 Grandparenthood

When do female members of the first generation become grandparents? It depends on the ages at birth of their children and grandchildren. The ages of women of generation 1 at birth of their children are the dates of birth of their children minus the dates of birth of the women:

```
idego <- subset (dLH$ID,dLH$gen==1 & dLH$sex=="Female")
ages <- Db(IDch(idego))-Db(IDmother(IDch(idego)))
```

The mean age is 29.92 years and the standard deviation is 6.02 years.

The ages of women in the first generation at birth of their grandchildren are the dates of birth of the grandchildren minus the dates of birth of the women:

```
idego <- dLH$ID[dLH$gen==1 & dLH$sex=="Female"]
# IDs of grandchildren
idgch <- IDch(IDch(idego))
# Ages if egos at birth of grandchildren
ages <- Db(idgch) - Db(IDmother(IDmother(idgch)))
```

The mean age of women at birth of grandchildren is 59.89 years and the standard deviation is 8.49 years.

The age at grandmotherhood is the age at birth of the first grandchild. Ages at grandmotherhood are computed in a number of steps:

```
# Determine,, for each ego, the ID of grandchildren
z=IDch(IDch(idego,keep_ego=TRUE),keep_ego=TRUE)
# IDs of grandchildren
idgch <- z$Child
# IDs of the grandmothers
idgm_gch <- z$Maternalgm
# Add dates of birth of grandchildren
z$db_gch <- Db(idgch)
# Is grandmother alive at birth of the grandchild?
z$alive <- Dd(idgm_gch) > Db(idgch)
# Table of number of grandmother alive (=TRUE) or dead (=FALSE) at birth of grandchildren
table (z$alive)
#>
#> FALSE  TRUE
#>    70   597
# Age at grandmotherhood
zz <- aggregate(z,by=list(z$Maternalgm),function(x) min(x))
# For each grandmother, the ID of first grandchild is
```

```
idgch1 <- zz$Child
# The age of grandmother at birth of first grandchild
agegm_gch1 <- Db(idgch1) - Db(IDmother(IDmother(idgch1)))
# Grandmother is alive at birth of first grandchild if zz$alive is 1
table (zz$alive)
#>
#>   0   1
#>  34 175
```

with alive a logical variable, which is TRUE if a woman is alive and FALSE otherwise.

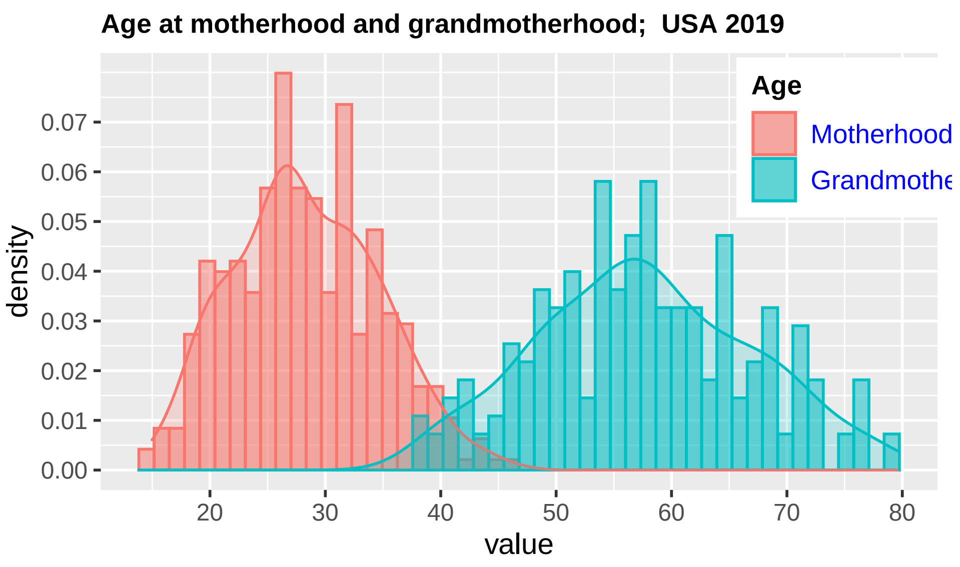
The number of first grandchildren to women in the first generation is 209. The proportion of woman in the first generation becoming a grandmother is 44.95 percent. Of the 209 women in generation 1 with at least one grandchild, 153 are alive at birth of their first grandchild.

The average age at grandmotherhood is 57.61 years and the standard deviation is 9.25 years. The age distributions at grandmotherhood is shown in the next figure. The figure also shows the ages at motherhood.

```
# ID of females in first generation with children
idego <- dLH$ID[dLH$gen==1 & dLH$sex=="Female" & dLH$nch>=1]
# ID of first child
idch <- IDch(idego)
idch1 <- idch[dLH$jch[idch]==1]
# ID of first grandchild
idgch <- IDch(IDch(idego))
zz <- data.frame(id=IDmother(IDmother(idgch)),idgch=idgch)
idgch1 <- aggregate(zz,by=list(zz$id),function(x) min(x))
# Create data frame with ID of mother, ID of first child and ID of first grandchild
d <- data.frame(idego=idego,idch1=idch1)
d$idgch1 <- NA
d$idgch1[idego%in%idgch1$id] <- idgch1$idgch
# Add age of ego at birth of first child
d$agem_ch1 <- Db(d$idch1) - Db(IDmother(d$idch1))
# Add age of ego at birth of first grandchild
d$agegm_gch1 <- NA
z1<- Db(d$idgch1)[!is.na(d$idgch1)]
z2 <- Db(IDmother(IDmother(d$idgch1)))
z3 <- z2[!is.na(d$idgch1)]
d$agegm_gch1[!is.na(d$idgch1)]  <- z1-z3
```

```
colnames(d)[4:5] <- c("Motherhood","Grandmotherhood")
Plot_agesgm <- function (agem_ch1,agegm_gch1)
{ # Age of mother and grandmother at birth of child
    dd <- reshape::melt.data.frame(d,id.vars="idego",measure.vars=c("Motherhood","Grandmotherhood"))
  colnames(dd)[2] <- "Age"
  ddd <- dd[!is.na(dd$value),]
  xmin <- 10
  xmax <- 90
  p <- ggplot(ddd, aes(x=value,color=Age,fill=Age)) +
              geom_histogram(aes(y=..density..), alpha=0.5, position="identity",bins=50)+
              geom_density(alpha=0.2) +
            scale_x_continuous(breaks=seq(xmin,xmax,by=10)) +
            scale_y_continuous (breaks=seq(0,0.07,by=0.01))
  # Add median
  p <- p + theme(legend.position=c(0.76,0.99),legend.justification=c(0,1)) +
    theme(legend.title = element_text(colour="black", size=10,face="bold")) +
    theme(legend.text = element_text(colour="blue", size=10,face="plain"))
  title <- paste ("Age at motherhood and grandmotherhood; ",attr(dLH,"country"),attr(dLH,"year") )
  p <- p + ggtitle(title) +
  theme(plot.title = element_text(size = 10, face = "bold"))
 p
}

Plot_agesgm(agem_ch1,agegm_gch1)
```



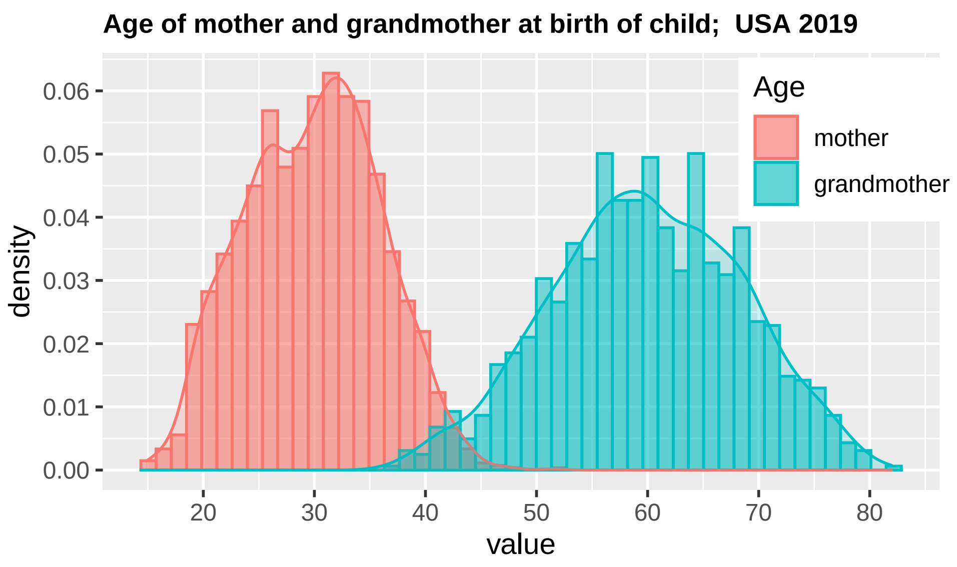
The duration of grandmotherhood is

```
yrs_gm <- ((Dd(d$idego)-Db(d$idego))-d$agegm_gch1)[!is.na(d$agegm_gch1)]
```

The mean duration is NaN years and the standard deviation NA years.

The following figure shows the ages of mothers and grandmothers at birth of a child or grandchild for all children (not only first children).

```
Plot_agesgm2 <- function (dLH)
{ # Age of mother and grandmother at birth of child
  id_ego <- dLH$ID
  agem <- dLH$bdated[id_ego]-dLH$bdated[IDmother(id_ego)]
  agegm <- dLH$bdated[id_ego]-dLH$bdated[IDmother(IDmother(id_ego))]
  d <- data.frame(id_ego=id_ego,mother=agem,grandmother=agegm)
  dd <- reshape::melt.data.frame(d,id.vars="id_ego",measure.vars=c("mother","grandmother"))
  colnames(dd)[2] <- "Age"
  xmin <- 10
  xmax <- 90
  p <- ggplot(dd, aes(x=value,color=Age,fill=Age)) +
             geom_histogram(aes(y=..density..), alpha=0.5, position="identity",bins=50)+
             geom_density(alpha=0.2) +
            scale_x_continuous(breaks=seq(xmin,xmax,by=10)) +
            scale_y_continuous (breaks=seq(0,0.07,by=0.01))
  # Add median
  p <- p + theme(legend.position=c(0.76,0.99),legend.justification=c(0,1))
  title <- paste ("Age of mother and grandmother at birth of child; ",attr(dLH,"country"),attr(dLH,"yea
  p <- p + ggtitle(title) +
  theme(plot.title = element_text(size = 10, face = "bold"))
 p
}
Plot_agesgm2 (dLH)
```



Let's now adopt the perspective of children on grandparenthood. The percentage of children in generation 3 with grandmother alive at time of birth is:

```
idego <- dLH$ID[dLH$gen==3]
# Grandmother of idego
idgm <- IDmother(IDmother(idego))
# Number of grandmothers alive at birth of egos
ngch3 <- length(idego[Dd(idgm)>=Db(idego)])
# Proportion
round (100*ngch3/length(idego),2)
#> [1] 89.51
```

For a child (reference person) to be able to celebrate a grandother's birthday, 85th birthday say, both the reference person and the grandmother should be alive. First consider a single member of the 3th generation. The entire third generation is considered next. Suppose ego is selected at random from the members of the third generation:

```
idego <- sample (dLH$ID[dLH$gen==3],1)
```

The grandmother of ego is

```r
idgm <- IDmother(IDmother(idego))
```

The grandmother is alive at age 85 if the date of birth of the grandmother is larger that the date of birth of ego plus 85.

```r
alive85 <- Dd(idgm)>=Db(idgm)+85
```

$alive85$ is a logical variable. It is true if the grandmother is alive and false if the grandmother is not alive at birth of ego.

The age of ego at the 85th birthday of the grandmother if alive is the decimal date of death of the grandmother minus the date of birth of the grandchild plus 85:

```r
agegch85 <- NA
agegch85[alive85] <- (Db(idgm)+85-Db(idego))[alive85]
```

If the grandmother is alive at age 85, $agegch85$ gives the age of the grandchild at the 85th birthday of the grandmother. If the grandmother is not alive at 85, $agegch85 = NA$.

The IDs of members of the third generation able to celebrate the 85th birthday of their living grandmother are:

```r
# IDs of members of third generation
idego <- dLH$ID[dLH$gen==3]
# IDs of their grandmothers
idgm <- IDmother(IDmother(idego))
# Is grandmother alive at 85?
alive <- Dd(idgm)>=Db(idgm)+85
# IDs of egos with living grandmother aged 85
idego85 <- idego[alive]
```

In the virtual population, 50.07 percent of the individuals are able to celebrate the 85th birthday of their grandmother.

The ages of these grandchildren at the 85th birthday of their grandmothers are:

```r
agegch85 <- Db(idgm[alive],dLH)+85-Db(idego85,dLH)
```

The mean age is 25.49 years and the standard deviation is 8.67 years. The proportion of grandchildren younger than 15 at grandmother's 85th birthday is 12.28 percent.

The age of ego (grandchild) at grandmother's death is computed in two steps. First, the age of grandmother at death is computed:

```r
agesgmd <- Dd(unique(idgm))-Db(unique(idgm))
```

Next, the age of ego at that event is computed:

```r
agesgchd <- Dd(IDmother(IDmother(idego)))-Db(idego)
```

The mean age is 22.06 and the standard deviation is 16.46. The figures depend on (a) the mean lifespan of women of the first generation who ever become a grandmother, (b) the mean age at childbearing of the women in the second generation with at least one child, and (c) the mean age at childbearing of women in the first generation with at least one child.

6 Double burden: young children and aging parent

Mothers with young children (less than 15) and at least one elderly parent (over 85) have a double burden, i.e. care for a young child and an elderly parent. Suppose the elderly parent is female. To determine the IDs of women with a double burden, we may adopt the woman's perspective or the child's perspective. A child aged less than 15 with a living mother and a living grandmother of 85 or older has a mother with a double burden. The IDs of mothers of children aged less than 15 and with a grandmother over 85 are:

```
iddb <-  unique(IDmother(idego85[agegch85<15]))
```

with *idego*85 the IDs of children with a grandmother of 85 or older and *idego*85[< 15] the IDs of those younger than 15.

The number of women of the second generation who experience a double burden during their lifetime is 33,which is 8.97 percent of the women of the second generation.

The number of years a mother experiences a double burden (from a child's perspective) is the number of years the grandmother lives beyond age 85 and the child is less than 15. a mother aged 85+ is the date of death of the egos' mothers minus the date of the 85th birthday of the mother

```
# IDs of mothers with double burden: iddb
# IDs of egos with mother who experiences double burden somewhere during her lifetime
idego_db <- IDch(iddb)
# Date of death of grandmother of ego with grandmother of 85+
dgmd <- Dd(IDmother(IDmother(idego_db)))
# 85th birthday of grandmother
dgm85 <- Db(IDmother(IDmother(idego_db))) + 85
# Date of birth of ego
dego0 <- Db(idego_db)
# 15th birthday of ego, assuming ego is alive at 15
dego15 <- Db(idego_db) + 15
# Length of episodes during which a child experiences a mother with double burden
period_db <- pmin(dgmd,dego15) - pmax(dgm85,dego0)
period_db[period_db<=0] <- NA
# Mean length of episode during which child has mother with double burden
mean(period_db,na.rm=TRUE)
#> [1] 2.947488
# Number of children with a mother going through episode of double burden
length(period_db[!is.na(period_db)])
#> [1] 41
# Women who ever experience a double burden have a total number of children of: length(idego_db)
# Some of these children contribute to the bouble burden, other do not
# ID is childen who contribute to mother's double burden
idego_db2 <- idego_db[!is.na(period_db)]
# ID of children with mothers with double burden who do not contribute to double burden
idego_db3 <- idego_db[is.na(period_db)]
# IDs of women with double burden and IDs of children contributing to double burden
idmch_db <- cbind (IDmother(idego_db2),idego_db2)
# Number of children contributing to double burden for each women with double burden
z <- aggregate(idmch_db,list(IDmother(idego_db2)),function(x) length(x))
addmargins(table (z[,3]))
#> 
#>   1   2   3 Sum 
#>  27   4   2  33
```

```
# IDs of women with highest number of children contributing to double burden
idm_max <- z$Group.1[z$V1==max(z$V1)]
```

A child who experiences an episode during which the mother has a double burden, has that experience for 2.95 years on average. The standard deviation is 2.66 years.

Alburez et al. (2021) consider individuals with at least one child aged 15 or younger and a parent or parent in-law within 5 years of death to be sandwiched. Applying that criterion, the following code computes the proportion of women with a double burden at some time during their lives:

```
# ID of individuals in generation 3 (children)
idego <- dLH$ID[dLH$gen==3]
# Ages of grandchildren five years prior to death of maternal grandmother
agegch5 <- Dd(IDmother(IDmother(idego)))-5-Db(idego)
# Is grandchild less than 15 five years prior to death of maternal grandmother?
chless15 <- agegch5<15 # Logical variable: true or false
# Number of grandchildren less than 15 five years prior to death of mother
table (chless15)
#> chless15
#> FALSE  TRUE
#>   397   270
# The mothers of these grandchildren have some episode of double burden
idmAll <-  unique(IDmother(idego))
idmdb <- unique(IDmother(idego[chless15])) # 396 mothers with 621 children < 15
# Proportion of women in generation 2 with double burden
z <- round (100*length(idmdb)/length(idmAll),2)
```

Of the women in generation 2 with children, 51.39 percent experience an episode of double burden. It is the size of "sandwich generation".

7 The fertility table: multistate life table

The fertility table consists of *expected values* of fertility indicators computed from conditional fertility rates (occurrence-exposure rates). The fertility indicators are:

- Multistate survival function (S): state occupation probabilities predicted from empirical transition rates. A state probability is the probability that an individual is in a given state at age x. It depends on (a), survival and (b) being in a state conditional on survival.
- Transition probabilities (P): probabilities that an individual in a given state at age x is in a given other state at age x+1.
- Expected state occupation times(L): expected time spent in a given state (expected sojourn time) between ages x and x+1. Expected state occupation times may be conditional on the state occupied at age x, conditional on survival but not on the state occupied at x, or unconditional on survival and state occupied at x. The first measure is "status-based", the second and third measures are is "population-based".
- Expected state occupation times beyond a given age
  - Expected state occupation times at birth (e0).
  - Expected state occupation times at age x, by status occupied at age x
  - Expected state occupation times at age x, irrespective of state occupied at x.

For the methodology of multistate life table construction, see the vignette *Simulation of life histories* included in the 'VirtualPop' package and Willekens (2014). A fertility table is an illustration of the hierarchical multistate model, characterized by having r living states connected by r-1 rates of transition (Schoen 2016; Schoen 2019).

The expected state occupation times at birth are

```
z <- Multistate(rates)$mslt$e0[,1]
round(z,2)
#>     0     1     2     3     4     5
#> 33.63  7.44  7.49  3.49  1.18  0.75
```

A newborn girl (in generation 1) may expect to spend 33.6 years before entering motherhood. The figure is higher than the age at motherhood for those entering motherhood because a considerable proportion of the cohort never enters motherhood (and remain childless throughout the reproductive period). Women with one child spend an average of 7.4 years with one child before death or the end of the reproductive period, whatever comes first. Note that the sum of the lengths of episodes is less than 55, the end of the reproductive period, because of death before age 55.

References

Alburez-Gutierrez, D., Mason, C., & Zagheni, E. (2021). The "sandwich generation" revisited: Global demographic drivers of care time demands. *Population and Development Review*, *47*(4), 997–1023.

Arpino, B., Gumà, J., & Julià, A. (2018). Family histories and the demography of grandparenthood. *Demographic Research*, *39*(42), 1105–1150. https://doi.org/10.4054/DemRes.2018.39.42

Belanger, A., & Sabourin, P. (2017). *Microsimulation and population dynamics. An introduction to Modgen 12*. Springer. https://doi.org/10.1007/978-3-319-44663-9

Bumpass, L., & Lu, H.-H. (2000). Trends in cohabitation and implications for children's family contexts in the United States. *Population studies*, *54*(1), 29–41.

Lam, D., & Marteleto, L. (2008). Stages of the demographic transition from a child's perspective: Family size, cohort size, and children's resources. *Population and Development Deview*, *34*(2), 225–252.

Margolis, R. (2016). The changing demography of grandparenthood. *Journal of Marriage and Family*, *78*(3), 610–622.

Margolis, R., & Verdery, A. M. (2019). A cohort perspective on the demography of grandparenthood: Past, present, and future changes in race and sex disparities in the united states. *Demography*, *56*(4), 1495–1518.

Mills, M. C., & Rahal, C. (2021). Population studies at 75 years: An empirical review. *Population Studies*, *75*(sup1), 7–25.

Schoen, R. (2016). Hierarchical multistate models from population data: An application to parity statuses. *PeerJ*, *4*, e2535.

Schoen, R. (Ed.). (2019). *Analytical family demography* (Vol. 47). Springer.

Verdery, A. M. (2015). Links between demographic and kinship transitions. *Population and Development Review*, *41*(3), 465–484.

Wachter, K. W., Blackwell, D., & Hammel, E. A. (1997). Testing the validity of kinship microsimulation. *Mathematical and Computer Modelling*, *26*(6), 89–104. https://doi.org/10.1016/S0895-7177(97)00172-6

Willekens, F. (2009). Continuous-time microsimulation in longitudinal analysis. In A. Zaidi, A. Harding, & P. Williamson (Eds.), *New frontiers in microsimulation modelling* (pp. 413–436). Routledge. https://doi.org/10.4324/9781315248066

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

Wolf, D. A. (1994). The elderly and their kin: Patterns of availability and access. In L. Martin & S. Preston (Eds.), *Demography of aging* (pp. 146–194). Washington, DC: National Academy Press.