Families and Kinship Networks in a Virtual Population. The *Families* package.

Frans Willekens

2022-06-10

Introduction

A multi-generation virtual population offers an opportunity to study family ties and kinship networks. Individuals with social ties constitute a social network. A *holistic perspective* on a network concentrates on network characteristics. An *egocentric* perspective 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 in a four-generation population, 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. 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.

A few studies on the three subjects listed exist. The studies predominantly use microsimulation and SOCSIM is the most popular software package. SOCSIM creates individual life histories and genealogies from age-specific mortality and fertility rates (Wachter et al. 1997). The microsimulation method is similar to the one used in the *VirtualPop* package to generate a virtual population. In *VirtualPop* simulation proceeds in continuous time[1] Microsimulation in continuous time has two important advantages (Willekens 2017). 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. Second, the exact duration between events can be computed. 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. The perspective of children is rarely adopted in demography (Mills and Rahal 2021, p. 20). A child interprets demography as the presence of parents, grandparents, siblings and peers. 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 double burden was recently studied by Alburez-Gutierrez et al. (2021). The authors uses microsimulation to study the double burden in countries around the world. The input data consist of mortality and fertility rates of 198 countries and territories included in the 2019 Revision of the UNWPP. The input data cover the period 1950-2100. 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."

Note that the life histories of members of the virtual population produced by *VirtualPop* are not projections

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

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 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 in R.

This document consists of three sections in addition to the introduction. The next section consists of five subsections. The first Is a very brief introduction to *VirtualPop*. The second presents six basic queries. They are sufficient to produce kinship networks. The other subsections illustrate the use of the queries to obtain information on the kin of ego: siblings, cousins and aunts. The third 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. The last section concentrates on the double burden. It addresses questions such as age at onset and termination, and the proporition of people who experience a double burden.

Retrieve family ties

The *VirtualPop* package

The *VirtualPop* package generates a virtual population of given size and number of generations from death rates by age (single years of age) and sex, and fertility rates by age and birth order. Rates may be downloaded from the Human Mortality Database (HMD) and the Human Fertility Database (HFD). In the HFD, the rates *VirtualPop* uses are called *conditional fertility rates*. For details, see the vignettes included in the *VirtualPop*.

The fertility and mortality rates of the United States in the calendar year 2019 and the virtual population generated from these rates are included in the data folder of the *Families* package. To load the virtual population, the following code is used:

```
library (Families)
data(dataLH)
dLH <- dataLH
ngen <- max(unique (dLH$gen))
```

The virtual population consists of 30151 individuals belonging to 4 generations. The following table shows the number of individuals by generation and sex:

```
addmargins(table (Generation=dataLH$gen,Sex=dataLH$sex))
```

```
##           Sex
## Generation  Male Female   Sum
##        1    5029   4971 10000
##        2    4128   4121  8249
##        3    3241   3365  6606
##        4    2614   2682  5296
##        Sum 15012  15139 30151
```

Basic functions

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

A seventh function is added for convenience:

- $IDgch(idego)$ retrieves the ID of ego's grandchildren

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: $Db(idego) - Dd(idego)$.

The functions may be used in combination. 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.

By way of illustration, let's retrieve the data on three individuals and on their maternal ancestors. The individuals are ID 270923, 211892 and 227091. The retrieval of information on ancestors requires (a) retrieval of the IDs of the ancestors and (b) retrieval of the individual records of these persons. Calling the function $IDmother()$ does not guarantee that the desired function is called. The Comprehensive R Archive Network (CRAN) has about 20 thousand contributed packages. Some of these packages may have function called $IDmother$. To ensure that the correct function is accessed, the double colon operator is used. The second line of the code chunk that follows ensures that $IDmother$ accessed the function by that name in the package $Families$ and not in any other package.

```r
idego <- c(27023,21192,22091)
IDmother <- Families::IDmother
dataLH[c(idego,IDmother(idego,dLH),IDmother(IDmother(idego,dLH),dLH)),]
```

```
##          ID gen    sex   bdated   ddated      x_D IDpartner IDmother IDfather
## 27023 27023   4   Male 2109.480 2190.310 80.83073     26399    20964    23418
## 21192 21192   3   Male 2083.063 2170.194 87.13041     18516    13627    12671
## 22091 22091   3 Female 2087.941 2174.731 86.79000     20409    14680    16787
## 20964 20964   3 Female 2086.981 2164.006 77.02485     24399    13380    16674
## 13627 13627   2 Female 2050.956 2130.608 79.65232     10266     4359     8614
## 14680 14680   2 Female 2052.817 2137.734 84.91654     13805     5710     6490
## 13380 13380   2 Female 2056.450 2144.801 88.35127     10503     4058     5150
## 4359   4359   1 Female 2019.580 2099.901 80.32104      3237       NA       NA
## 5710   5710   1 Female 2019.683 2105.420 85.73782      5178       NA       NA
##       jch nch  id.1  id.2  id.3  id.4  id.5 id.6 id.7 id.8 id.9    age.1
## 27023   2   5 31477 31478 31479 31480 31481   NA   NA   NA   NA       NA
## 21192   1   3 25065 25066 25067    NA    NA   NA   NA   NA   NA       NA
## 22091   1   0    NA    NA    NA    NA    NA   NA   NA   NA   NA       NA
## 20964   1   5 27022 27023 27024 27025 27026   NA   NA   NA   NA 21.88270
## 13627   4   2 21192 21193    NA    NA    NA   NA   NA   NA   NA 32.10764
## 14680   2   2 22091 22092    NA    NA    NA   NA   NA   NA   NA 35.12382
## 13380   1   2 20964 20965    NA    NA    NA   NA   NA   NA   NA 30.53106
## 4359   NA   4 13624 13625 13626 13627    NA   NA   NA   NA   NA 25.56604
## 5710   NA   2 14679 14680    NA    NA    NA   NA   NA   NA   NA 24.35204
##           age.2    age.3    age.4    age.5 age.6 age.7 age.8 age.9
## 27023       NA       NA       NA       NA    NA    NA    NA    NA
## 21192       NA       NA       NA       NA    NA    NA    NA    NA
## 22091       NA       NA       NA       NA    NA    NA    NA    NA
## 20964 22.49865 24.72482 26.36746 36.04109    NA    NA    NA    NA
## 13627 32.97360       NA       NA       NA    NA    NA    NA    NA
## 14680 35.14795       NA       NA       NA    NA    NA    NA    NA
## 13380 32.87715       NA       NA       NA    NA    NA    NA    NA
## 4359  30.14332 31.32168 31.37614       NA    NA    NA    NA    NA
## 5710  33.13458       NA       NA       NA    NA    NA    NA    NA
```

For each individual selected, the table includes information on the individual, the ID of the partner, the ID of the mother and father (if their generation is included in the virtual population), the number of children during her lifetime (nch), the IDs of the children and the ages of the mother at childbearing. The first individual has 5 children. The IDs of the children are $rdataLH[27023,(12:(11+dataLH\mathrm{nch}[27023]))]$‘. To list the IDs of egos and ancestors, the function call is:

```
IDmother(IDmother(idego,dLH,keep_ego=TRUE),dLH)
```

```
##    IDgm IDmother IDego
## 1 13380    20964 27023
## 2  4359    13627 21192
## 3  5710    14680 22091
```

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 $IDmother$. The variable IDgm denotes the ID of the grandmother of ego.

The IDs of the women in generation 1 of the virtual population entering motherhood are

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

with idm the IDs of women with children. The IDs of women without children are $idego[idego$. The proportion remaining childless is $1 - length(idm)/length(idego)$, which gives 23.42 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
## 1164 1174 1507  700  238  119   69 4971
```

The IDs of the grandchildren of individual with ID 2 is

```
IDgch(id=2,dataLH=dLH)
```

```
##  [1] 18748 18749 18937 18938 18939 19320 20887 20888 20889 20985 20986 20987
```

For each woman in the generation 1, the ID of the oldest child is (for each ego, select child with lowest date of birth)

```
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
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     2              10001                      2041.345
## 2     3              10007                      2055.860
## 3     5              10009                      2049.017
## 4     7              10010                      2052.817
## 5     8              10012                      2056.241
## 6    11              10013                      2048.505
```

For each woman, the ID of the youngest child is

```
# Select, for each ego, child with highest date of birth
ch_youngest=aggregate(zz,list(dLH$IDmother[idch]),function(x) max(x))
colnames(ch_youngest) <- c("idego","ID of youngest child","date of birth of youngest child")
```

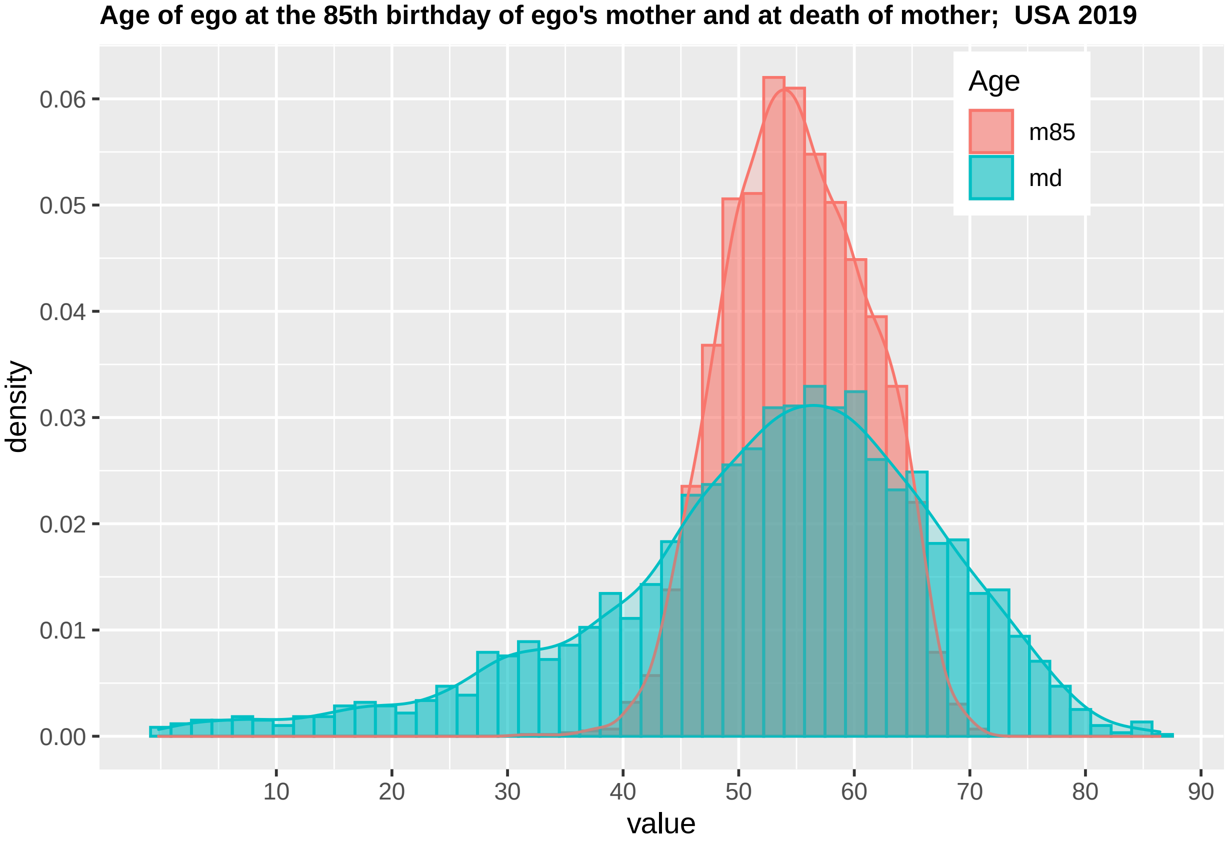
ch_youngest is a data frame with three columns: the ID of the mother, the ID of the youngest child and the date of birth of the youngest child.

Consider women of the third generation. The age of ego at the 85th birthday of ego's mother:

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

```
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))
  # 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 7.65 years and the standard deviation is 4.99 years.

The proportion of mothers alive at age 65 of egos is:

```
refage <- 65
# When does ego turn refage?
ego_refage <- dLH$bdated[idego] + refage
# Is mother alive at age 70 of ego?
alive <- dLH$ddated[IDmother(idego,dLH)] >=ego_refage
# Table of number of egos with mother alive (=TRUE) or dead (=FALSE)
table (alive)
```

```
## alive
## FALSE  TRUE
##  2713   652
```

```
# Age of living mothers at refage of ego
age_m <- (dLH$ddated[IDmother(idego,dLH)] - dLH$bdated[IDmother(idego,dLH)])[alive]
```

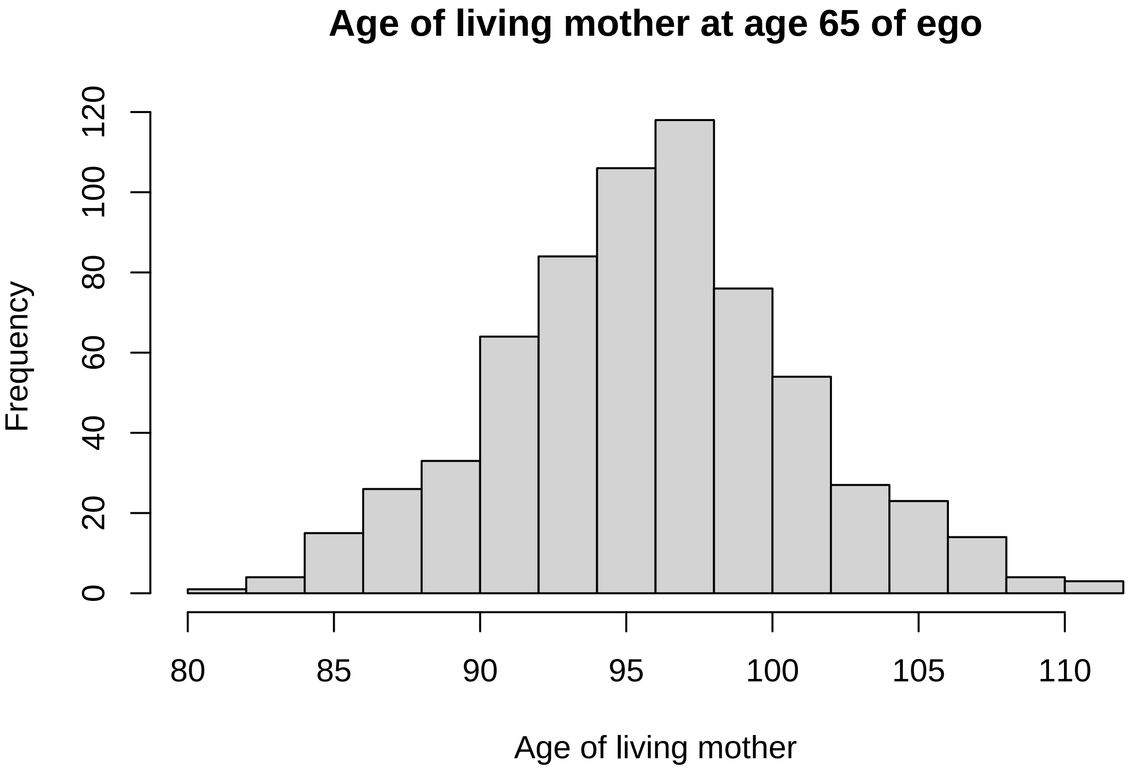
```
mean(age_m)
```

```
## [1] 95.84481
```

```
sd(age_m)
```

```
## [1] 5.060509
```

```
hist(age_m,main=paste ("Age of living mother at age ",refage," of ego",sep=""),breaks=20,xlab="Age of li
```



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 their partners. Consider the individual with idego=27023. The IDs of ego's siblings are retrieved in two steps. First, the IDs of the children of ego's mother are retrieves:

```
idego <- 27023
IDch(IDmother(idego,dLH),dLH)
```

```
## [1] 27022 27023 27024 27025 27026
```

Second the ID of ego is removed. The IDs of ego's siblings are:

```
IDch(IDmother(idego,dLH),dLH)[IDch(IDmother(idego,dLH),dLH)%in%idego==FALSE]
```

```
## [1] 27022 27024 27025 27026
```

The code uses subsetting, which is a feature of vectorized operations. Most of the functions in R are vectorized, which means that a function operates on all elements of a vector simultaneously. Vectorized operations prevent the loops that makes R slow[2].

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

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

```
idego <- c(27023,21192,22091)
IDch(IDmother(idego,dLH),dLH)
```

```
## [1] 21192 21193 22091 22092 27022 27023 27024 27025 27026
```

The siblings of the three unrelated reference persons are:

```
IDch(IDmother(idego,dLH),dLH)[IDch(IDmother(idego,dLH),dLH)%in%idego==FALSE]
```

```
## [1] 21193 22092 27022 27024 27025 27026
```

If two reference persons have the same mother, the approach does not work. For instance,

```
idego <- c(27023, 27025,21192,22091)
IDch(IDmother(idego,dLH),dLH)[IDch(IDmother(idego,dLH),dLH)%in%idego==FALSE]
```

```
## [1] 21193 22092 27022 27024 27026
```

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 members included in the list of reference persons. The reason is that reference persons are excluded from the list of siblings. The following code produces a list object of four elements, one for each reference person. A list element shows the IDs of siblings of the reference person.

```
f <- function(idego)
      y <- IDch(IDmother(idego,dLH),dLH)[IDch(IDmother(idego,dLH),dLH)%in%idego==FALSE]
kk <- list()
for (i in 1:length(idego))
{ z <- f(idego[i])
  kk[[i]] <- z
}
kk
```

```
## [[1]]
## [1] 27022 27024 27025 27026
##
## [[2]]
## [1] 27022 27023 27024 27026
##
## [[3]]
## [1] 21193
##
## [[4]]
## [1] 22092
```

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,dLH),dLH)[IDch(IDmother(idego,dLH),dLH)%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 27023 is

```
sib[[which(idego==10001)]]
```

```
## [1] 10002 10003 10004 10005 10006
```

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 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
## 14.23 36.54 25.46 11.54  7.21  5.02
```

```
mean <- mean(z,na.rm=TRUE)
sd <- sd(z,na.rm=TRUE)
```

The mean number of siblings is 1.7602134 and the standard deviation is 1.3164308.

The age distribution of siblings at age 10 of egos is:

```
# Total number of siblings (for all members of pop)
length(sib)
```

```
## [1] 8249
```

```
# Ages of siblings at age 10 of ego
m <- as.list(idego)
# Merge two lists
zz <- 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],dataLH)
        # Dates of birth of siblings
        db_sib <- Db(x[2:length(x)],dataLH)
        # 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)
  }
mage_sib <- AgeSib(zz,ageRef=10)
mage_sib0 <- AgeSib(zz,ageRef=0)
```

The mean age of siblings at age 10 of ego 10.98 years with a standard deviation of 5.48 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 5.15 years with standard deviation of 3.96 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.

A child's perspective on family composition differs significantly from the perspective of parents:

```r
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$nch[idego]))
# Proportion of motherrs by number of children ever born
zz <- round (100 * table (dLH$nch[idego])[2:6]/sum(table (dLH$nch[idego])[2:6]),2)
```

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. From a child's perspective, only children represent only 14 percent of all children. Only children are much less common from a child's perspective than from a mother's perspective.

Cousins

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

```r
idego <- 27023
idgm <- IDmother(IDmother(idego,dLH),dLH)
ch <- IDch(IDch(idgm,dLH),dLH)
idcousins <- ch[ch%in%IDch(IDmother(idego,dLH),dLH)==FALSE]
idcousins
```

```
## [1] 30125 30126
```

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 sample of three members of the third generation. The IDs of their cousins are:

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

To get each reference person's cousins, use

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

The code is used to identify the maternal cousins of every member of the third generation:

```r
idego <- dLH$ID[dLH$gen==3]
idgm <- IDmother(IDmother(idego,dLH),dLH)
cousins <- list()
```

```
for (i in 1:length(idego))
{ z <- f(idego[i],idgm[i],dLH)
  cousins[[i]] <- z
}
```

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

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

```
z <- sapply(cousins,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
## 24.33 12.56 19.01 11.43  9.85  6.31  5.57  3.72  2.35  1.82  1.07  1.12  0.29
##    13    14    15    16
##  0.21  0.20  0.12  0.03
```

One of of four children have no maternal cousins, while one out of five have 2 cousins.

The frequency distribution of parental cousins is

```
idego <- dLH$ID[dLH$gen==3]
idgm <- IDmother(IDfather(idego,dLH),dLH)
cousins <- list()
for (i in 1:length(idego))
{ z <- f(idego[i],idgm[i],dLH)
  cousins[[i]] <- z
}
z <- sapply(cousins,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
## 25.90 12.40 16.30 13.06  9.58  7.11  5.03  3.50  2.45  1.71  0.92  0.91  0.83
##    13    14    15    16
##  0.15  0.08  0.03  0.03
```

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

Aunts

Maternal aunts are sisters and sisters-in-law of the reference persons mother. The IDs are obtained in two steps. First, the IDs of the daughters and the IDs of 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)[dLH$sex[IDch(idgm,dLH)]=="Female"],
              IDpartner(IDch(idgm,dLH)[dLH$sex[IDch(idgm,dLH)]=="Male"],dLH))
              aunts <- iddaught[iddaught%in%IDmother(id,dLH)==FALSE]
            }
```

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

The frequency distribution of maternal aunts of members of the third generation 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
## 13.64 37.06 25.04 11.99  7.58  4.69
```

Grandparenthood

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 mothers:

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

The mean age is 29.98 years and the standard deviation is 5.99 years.

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

```
idego <- dLH$ID[dLH$gen==1 & dLH$sex=="Female"]
idgch <- IDch(IDch(idego,dLH),dLH)
ages <- dLH$bdated[idgch] - dLH$bdated[IDmother(IDmother(idgch,dLH),dLH)]
```

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

```
#IDs of first grandchildren
idgch1 <- idgch[dLH$jch[idgch]==1]
# IDs of the grandmothers
idgm_gch1 <- IDmother(IDmother(idgch1,dLH),dLH)
# Dates of birth of first grandchildren
db_gch1 <- dLH$bdated[idgch1]
# IDs of grandmothers alive at birth of the first grandchild
alive <- Dd(idgm_gch1,dLH) > Db(idgch1,dLH)
# Age at grandmotherhood
agegm_gch1 <- (Db(idgch1,dLH) - Db(idgm_gch1,dLH))[alive]
```

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

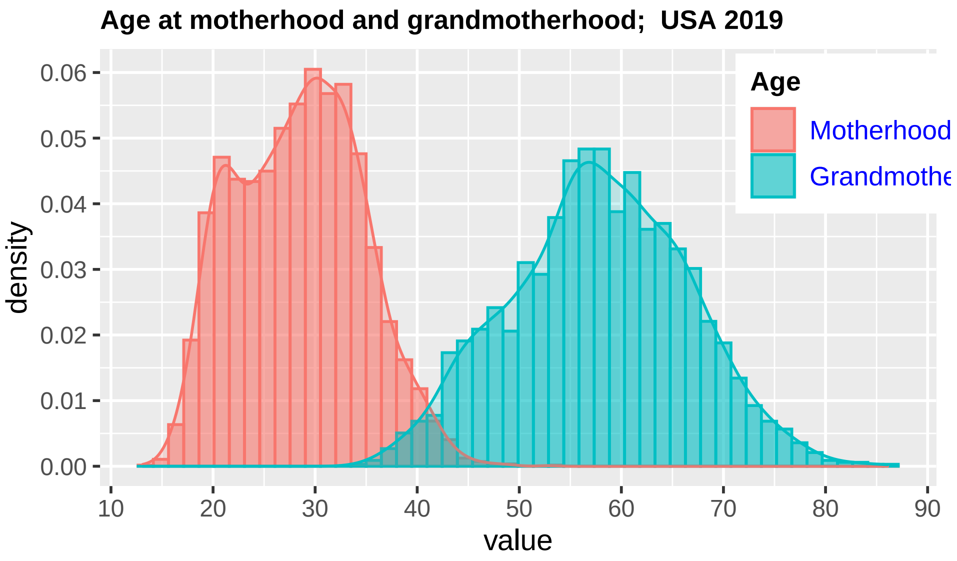
The number of first grandchildren to women in the first generation is 6170. The proportion of woman in the first generation becoming a grandmotherhood is 124.1198954 percent. About 5 percent do not survive until the birth of the first grandchild. Of the 6170 women in generation 1 with at least one grandchild, 5702 are alive at birth of their first grandchild.

The average age at grandmotherhood is 58.04 years and the standard deviation is 8.45 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,dLH)
idch1 <- idch[dLH$jch[idch]==1]
# ID of first grandchild
idgch <- IDch(IDch(idego,dLH),dLH)
zz <- data.frame(id=IDmother(IDmother(idgch,dLH),dLH),idgch=idgch)
idgch1 <- aggregate(zz,by=list(zz$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,dLH) - Db(IDmother(d$idch1,dLH),dLH)
# Add age of ego at birth of first grandchild
d$agegm_gch1 <- NA
z1<- Db(d$idgch1,dLH)[!is.na(d$idgch1)]
z2 <- Db(IDmother(IDmother(d$idgch1,dLH),dLH),dLH)
z3 <- z2[!is.na(d$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(idgm_gch1,dLH)-Db(idgm_gch1,dLH))[alive]-agegm_gch1
```

The mean number is 27.3896584 years and the standard deviation 12.660375 years.

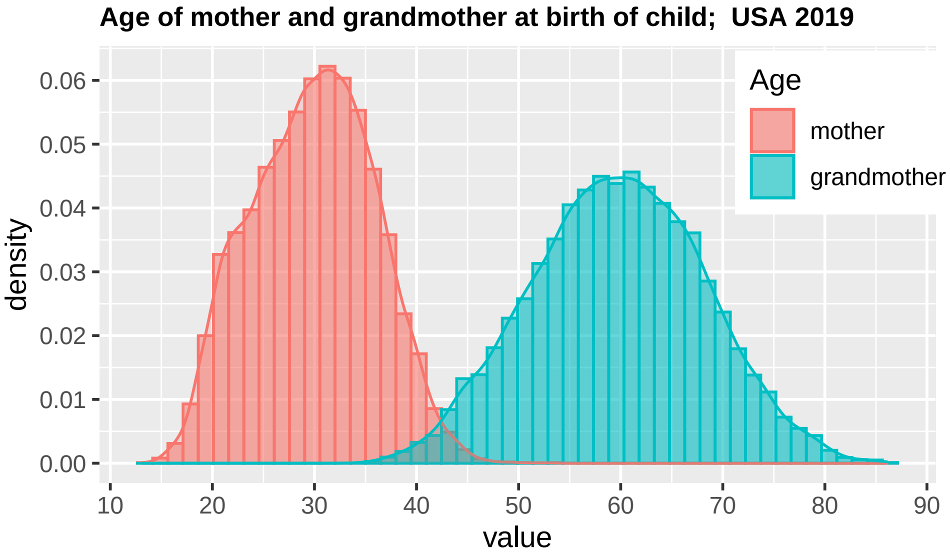
The percentage of children in generation 3 with grandparents alive at time of birth is:

```
idego <- dLH$ID[dLH$gen==3]
ngch3 <- length(idego[Dd(idgm,dLH)>=Db(idego,dLH)])
round (100*ngch3/length(idego),2)
```

```
## [1] 91.72
```

The following figure shows the ages of mothers and grandmothers at birth of a child or grandchild.

```
Plot_agesgm2 <- function (dLH)
{ # Age of mother and grandmother at birth of child
  id_ego <- dLH$ID
  agem <- dLH$bdated[id_ego]-dLH$bdated[IDmother(id_ego,dLH)]
  agegm <- dLH$bdated[id_ego]-dLH$bdated[IDmother(IDmother(id_ego,dLH),dLH)]
  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)
```



What proportion of members of the third generation are able to celebrate the 85th birthday of their grandmother? To celebrate the 85th birthday, reference persons and grandmothers should be alive. First consider a single member of the 3th generation, selected at random,

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

The grandmother of ego is

```
idgm <- IDmother(IDmother(idego,dLH),dLH)
```

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. 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.

```
alive85 <- Dd(idgm,dLH)>=Db(idgm,dLH)+85
# alive85 is a logical variable: true or false
agegch85 <- NA
agegch85[alive85] <- (Db(idgm,dLH)+85-Db(idego,dLH))[alive85]
```

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

The IDs of members of the third generation at the 85th birthday of their living grandmothers is idgch:

```
idego <- dLH$ID[dLH$gen==3]
idgm <- IDmother(IDmother(idego,dLH),dLH)
alive <- Dd(idgm,dLH)>=Db(idgm,dLH)+85
idego85 <- idego[alive]
```

The ages of these egos (grandchildren) are

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

In the virtual population, 52.09 percent of the individuals are able to celebrate the 85th birthday of their grandmother. It is also the proportion of grandmothers alive at 85. The mean age of grandchildren at their grandmother's 85th birthday is 25.0531663 years with standard deviation of 8.4032589 years. 11.42 percent of the grandchildren are younger than 15 years.

The age of a grandchil at grandmother's death is computes in two steps. First, the age of grandmother at death is computed:

```
agesgmd <- Dd(unique(idgm),dLH)-Db(unique(idgm),dLH)
```

Next, the age of grandchildren at that event are computed:

```
agesgchd <- Dd(IDmother(IDmother(idego,dLH),dLH),dLH)-Db(idego,dLH)
```

The mean age is 23.26 and the standard deviation is 15.86.

Double burden: young children and aging parent

Mothers with young children (less than 15) and at least one elderly grandparent (over 85) have a double burden, i.e. care for a young child and an elderly parent. Suppose the elderly parent is female. The IDs of these mothers with a double burden are

```
iddb <-  unique(IDmother(idego85[agegch<15],dLH))
```

0.05 percent of the women of the third generation with children under 15 experience a double burden during their lifetime.

The number of years with a mother aged 85+ the date of death of the egos' mothers minus the date of the 85th birthday of the mother

```
agemd <- dLH$ddated[IDmother(idego85,dLH)] -Db(idego85,dLH)
agem85 <- dLH$bdated[IDmother(idego85,dLH)] + 85 - Db(idego85,dLH)
nyrs85 <- agemd - agem85
nyrs85[nyrs85<0] <- NA
```

The mean number of years is 7.6 and the standard deviation is 4.92.

Note that the number of years with a mother aged 85 and over may also be computed by as the mother's remaining number of years at age 85:

```
z <-  dLH$ddated[IDmother(idego85,dLH)] - (dLH$bdated[IDmother(idego85,dLH)] + 85)
nyrs85 <- z[z>0]
```

The number of years with a child below 15 is the date of birth of the youngest child plus 15 (provided the child survives until age 15).

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 this 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
idego <- dLH$ID[dLH$gen==3]
# Ages of grandchildren five years prior to death of maternal grandmother
agegch5 <- Dd(IDmother(IDmother(idego,dLH),dLH),dLH)-5-Db(idego,dLH)
# 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
##  4179  2427
```

```
# The mothers of these grandchildren have some episode of double burden
idmAll <-  unique(IDmother(idego,dLH))
idmdb <- unique(IDmother(idego[chless15],dLH)) # 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, 45.8 percent experience an episode of double burden. It is the size of "sandwich generation".

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

Bélanger, 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.

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. (2017). 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

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.