# A generic microsimulation for modelling and projecting family structure and intergenerational relationships

Sabine Zinn

January 2024, Vienna

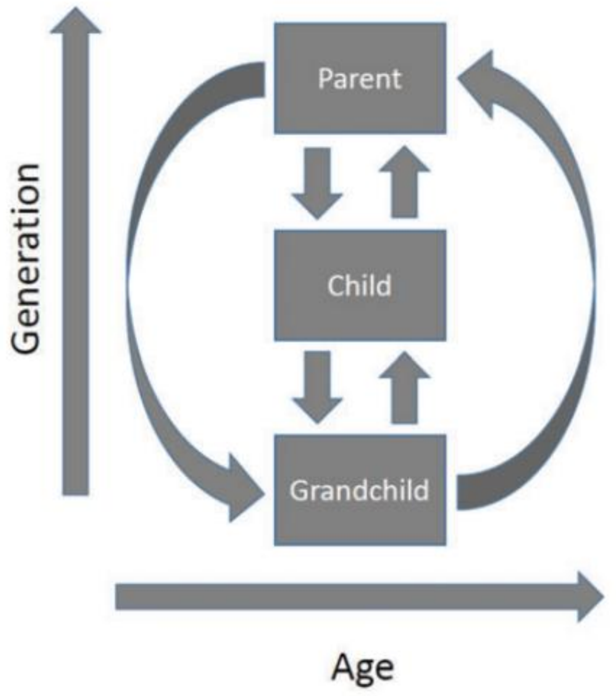
9th World Congress of the International Microsimulation Association

# Motivation

- Family context and genes matter for (almost) all societies worldwide
- Especially in social and economic contexts

- What if … things would change → Microsimulation approaches
- Software: There is SocSim[1] … but it is written in C and fixed to pre-determined demographic model structure
- Better: Have a ***generic open source, up-to-date, easy to enhance tool usable in a widely used software environment***:
  MicSim package in R and GitHub

FIGURE 3. LINKED LIVES AND CUMULATIVE INEQUALITY IN A MULTIGENERATIONAL PERSPECTIVE.



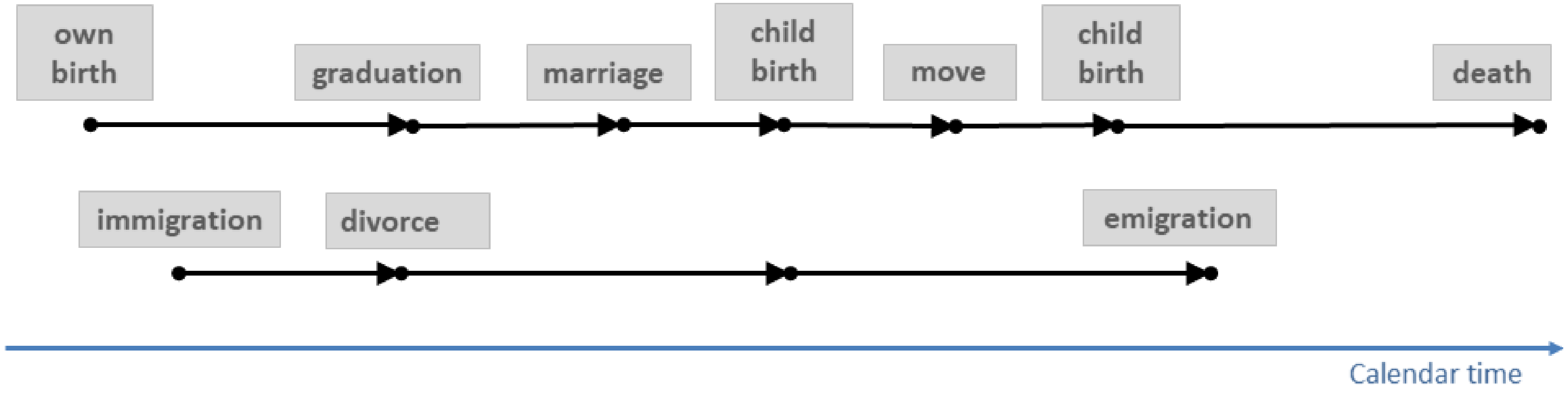

Gilligan, Karraker, & Jasper (2018). *Linked Lives and Cumulative Inequality: A Multigenerational Family Life Course Framework. Journal of Family Theory & Review, 10(1), 111–125.* doi:10.1111/jftr.12244

[1] https://lab.demog.berkeley.edu/socsim/

# Continuous Time Microsimulation along Calender Time

***Key idea***: Individuals 'live' their lives according to some (stochastic) model

***Discrete states***: summarize (demographically) relevant categories an individual can belong to (→ state space)
***Virtual population***: all individuals that are considered during simulation



# Life-Course Model: Continuous Time Markov Multi State Model

***Stochastic process*** that at any point in time occupies one out of a set of discrete states

***Key quantities***: transition rates $\lambda_{sj,sk}(t)$ of Markovian process $Z(t)$

Allow deriving ***distribution function*** $F(w_{sj}, t) \ = \ 1 - \ S(w_{sj}, t)$ ***of waiting time*** to next event

# Life-Course Model: Continuous Time Markov Multi State Model

***Stochastic process*** that at any point in time occupies one out of a set of discrete states

***Key quantities***: transition rates $\lambda_{sj,sk}(t)$ of Markovian process $Z(t)$

Allow deriving ***distribution function*** $F(w_{sj}, t) = 1 - S(w_{sj}, t)$ ***of waiting time*** to next event

The probability that an individual is still in state $s_j$ at time $t$ after waiting time $w_{sj}$ depends on all $K$ ***competing risks***

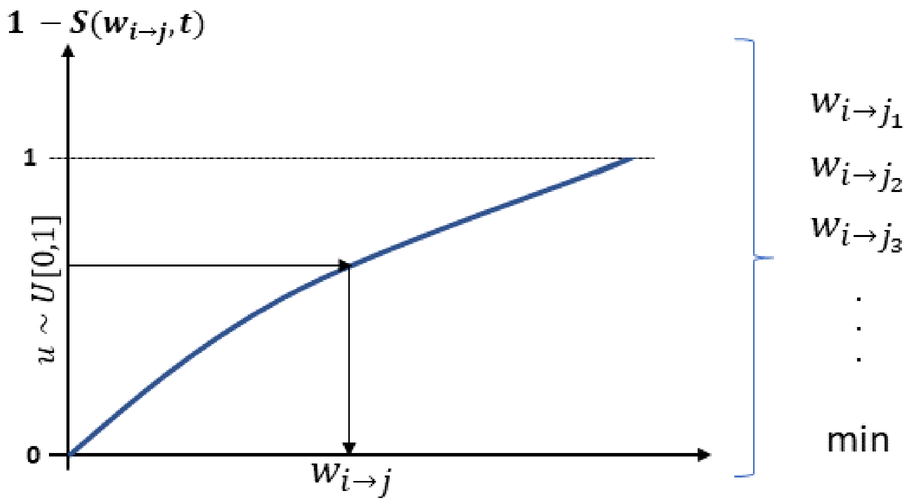
$$S(w_{sj}, t) = \prod_{k=1,k\neq j}^{K} \exp\left(-\Lambda(w_{sj,sk}, t)\right), \quad \Lambda\left(w_{sj,sk}, t\right) = \int_t^{w_{sj,sk}} \lambda_{sj,sk}(v)dv$$

with $w_{sj,sk}$ is the waiting time in $s_j$ after moving to $s_k$

# Data Ingredients

***Transition rates:***

- For each micro unit: Generate sequence of random waiting times to next events → **synthetic life-courses**
- Transition rates are estimated from survey data or vital statistics

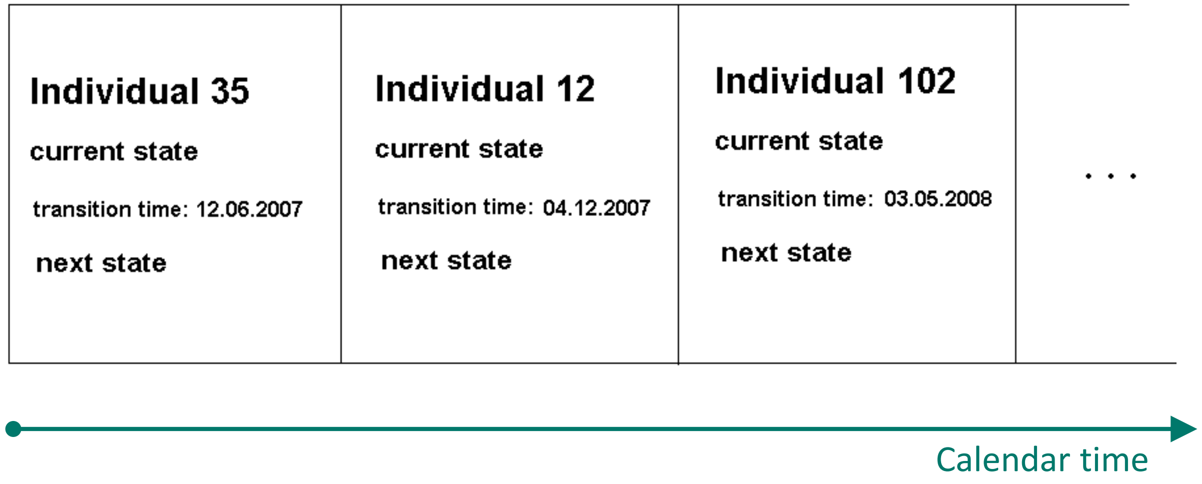



***Base population:*** Individuals to start with (e.g., census data)

# Simulation Processing

*At simulation starting time*

- For all members of base population next event times are computed
- These are inserted into a list & sorted according to time



# Simulation Processing

*At simulation starting time*

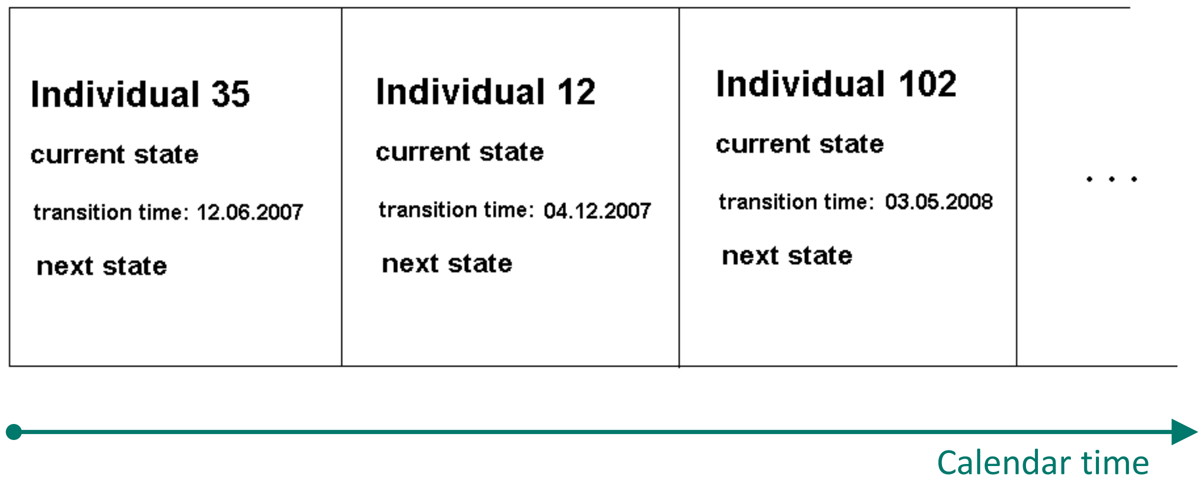
- For all members of base population next event times are computed
- These are inserted into a list & sorted according to time

*Repeat until simulation stopping time is reached*

- Dequeue first element from list
- Perform corresponding event
- Compute new event(s) for the respective unit(s) (possibly more events if more units are involved)
- Enqueue event(s) at the 'right' position

⇒ Virtual population evolves along calendar time

**⇒ Allows adding and simulating linked lives**



# An Example: Female Centered Model

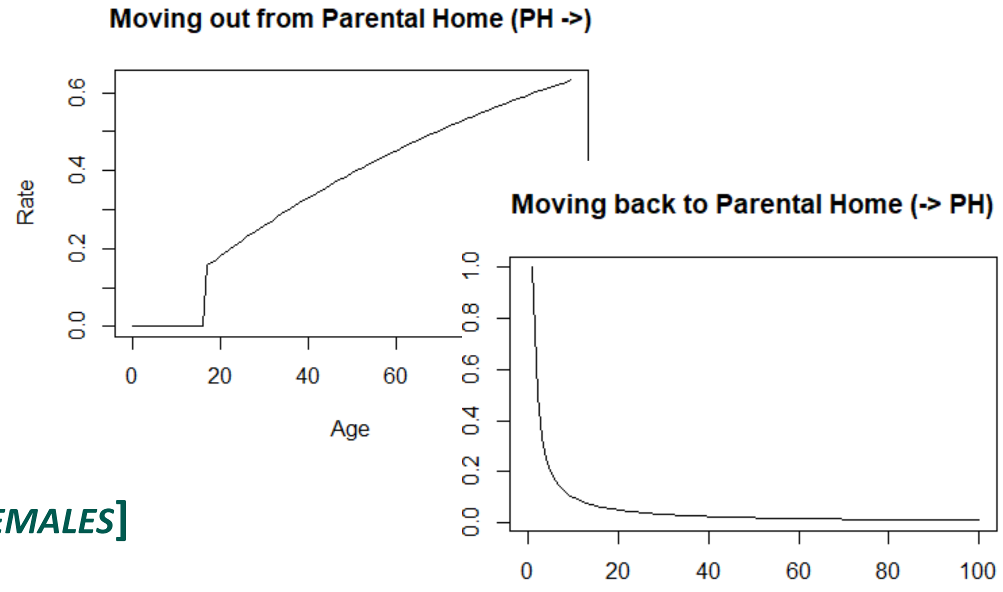
***Transitions***

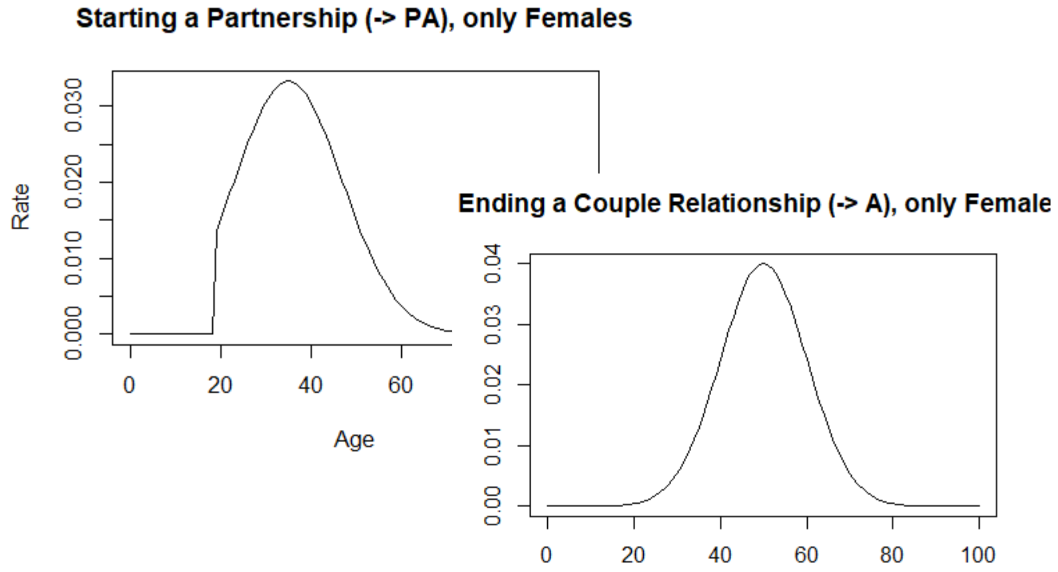
- Move out from parental home **[*PARENTAL HOME →„ALL“*]**
- Move back home **[*„ALL“→ PARENTAL HOME*]**
- Start partnership **[Parental Home / *ALONE → PARTNERSHIP; ONLY FEMALES*]**
- Stop partnership **[PARTNERSHIP→ ALONE; ONLY FEMALE]**
- Fertility **[CHILDLESS / CHILD→ CHILD(REN); ONLY FEMALE]**
- Mortality rates **[→ DEAD]**

In addition: migration (*optional*)

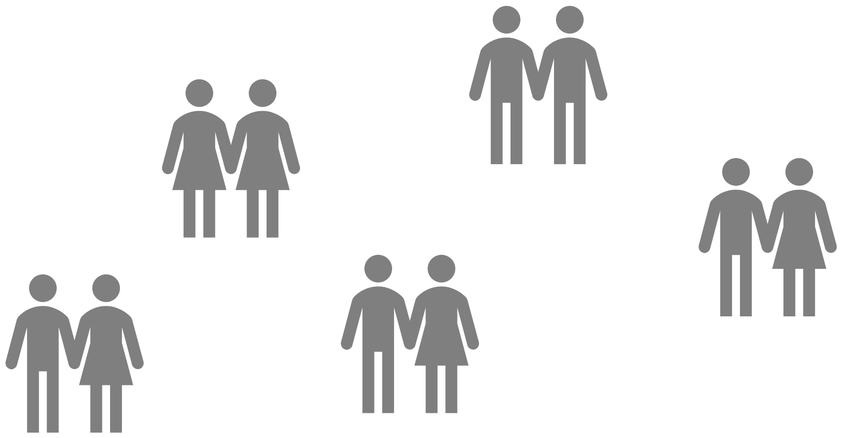
***Derived events***

- Widowhood for females and males
- Onset and quitting partnership for males
- Fertility for males



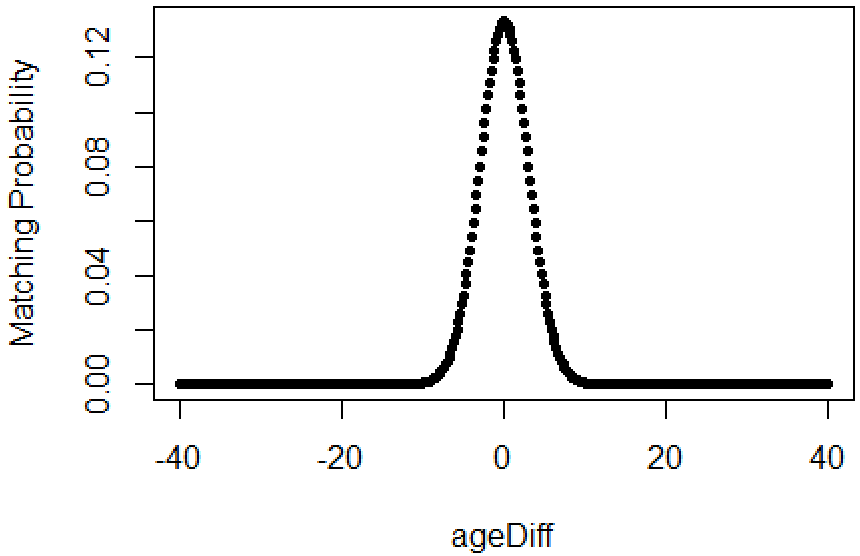



# Find a Partner



- Forming ***partnership market*** to choose from
- Following ***random procedure*** to assure that not only optimal matches are formed (reduce risk of unsuccessful searches)
- Letting choice rule depend on ***empirical patterns*** (e.g. partnership age profiles)

- Not allow incest
- Rule can be adapted by modeller (i.e. ***is generic***)



# Implemented in MicSim Package: micSimLink

```
# ------------------------------------------------------------------------------
# Defining simulation horizon
# ------------------------------------------------------------------------------
startDate <- 20140101 # yyyymmdd
endDate   <- 20641231 # yyyymmdd
simHorizon <- c(startDate=startDate, endDate=endDate)

# ------------------------------------------------------------------------------
# Seed for random number generator
# ------------------------------------------------------------------------------
set.seed(234)

# ------------------------------------------------------------------------------
# Definition of maximal age
# ------------------------------------------------------------------------------
maxAge <- 100

# ------------------------------------------------------------------------------
# Definition of non-absorbing and absorbing states
# ------------------------------------------------------------------------------
sex <- c("m","f")
livArr <- c("PH", "PA", "A")
fert <- c("0","1")
stateSpace <- expand.grid(sex=sex,livArr=livArr, fert=fert)
absStates <- "dead"
```

DIW SOEP

```r
# ----------------------------------------------------------------------
# Definition of (possible) transition rates
# ----------------------------------------------------------------------
# A. Moving out from parental home
moveOut <- function(age, calTime){
  return(ifelse(age>16,pexp((age)/mean(age), rate=0.5),0))
}
# B. Moving back to parental home
movePH <- function(age, calTime){
  rate <- 1/age
  return(rate)
}
# C. Starting partnership with living togehter
startPA <- function(age, calTime){
  rate <- dnorm(age, mean=35, sd=12)
  rate[age<=18] <- 0
  return(rate)
}
# D. Separation of partnership with living together
stopPA <- function(age, calTime){
  rate <- dnorm(age, mean=50, sd=10)
  return(rate)
}
# E. Fertility rates (Hadwiger mixture model)
fertRates <- function(age, calTime){
  b <- 3.5; c <- 28
  rate <-  (b/c)*(c/age)^(3/2)*exp(-b^2*(c/age+age/c-2))
  rate[age<=15 | age>=45] <- 0
  return(rate)
}
# F. Mortality rates (Gompertz model)
mortRates <- function(age, calTime){
  a <- .00003; b <- 0.1
  rate <- a*exp(b*age)
  return(rate)
}
```

```r
# ---------------------------------------------------------------------------------
# Define transition pattern
# ---------------------------------------------------------------------------------
partTrMatrix <- cbind(c("PH->A", "f/PH->f/PA", "f/A->f/PA", "f/PA->f/PH", "f/PA->f/A", "A->PH"),
                      c("moveOut", "startPA", "startPA","stopPA","stopPA", "movePH"))
fertTrMatrix <- cbind(c("f/0->f/1", "f/1->f/1"),c("fertRates","fertRates"))
allTransitions <- rbind(partTrMatrix,fertTrMatrix)
absTransitions <- cbind(c("f/dead", "m/dead"),
                        c(rep("mortRates",2)))

transitionMatrix <- buildTransitionMatrix(allTransitions=allTransitions,
                                          absTransitions=absTransitions,
                                          stateSpace=stateSpace)

# ---------------------------------------------------------------------------------
# Define transitions triggering a birth event
# ---------------------------------------------------------------------------------
fertTr <- fertTrMatrix[,1]

# ---------------------------------------------------------------------------------
# Define transitions triggering the onset of a partnership
# ---------------------------------------------------------------------------------
partTr <- c("PH->PA", "A->PA")
# Matching probability depends on age difference between potential partners
# with age difference defined as ageMale-ageFem
ageDiffDistr <- function(ageDiff) {
  return(dnorm(ageDiff, sd=3))
}

# ---------------------------------------------------------------------------------
# Define transitions triggering a separation
# ---------------------------------------------------------------------------------
sepTr <- c("PA->A", "PA->PH")
# Related occurrence probability for partners
probSepTr <- c(0.9, 0.1)

# ---------------------------------------------------------------------------------
# Define transitions in absorbing states (triggering also widowhood events)
# ---------------------------------------------------------------------------------
absPartTr <- c("dead -> A")
```

# Run it!

N=100.000 individuals

Over 50 years

Single core (Std): 1h 58min

```
# ---------------------------------------------------------------------------
# Execute microsimulation
# ---------------------------------------------------------------------------
pop <- micSimLink(initPop=initPop,
                  transitionMatrix=transitionMatrix, absStates=absStates,
                  varInitStates=varInitStates, initStatesProb=initStatesProb,
                  maxAge=maxAge, simHorizon=simHorizon,
                  fertTr=fertTr, partTr=partTr, rule=1, ageDiffDistr = ageDiffDistr,
                  sepTr=sepTr, probSepTr = probSepTr,
                  absPartTr=absPartTr,
                  duration=FALSE)
```

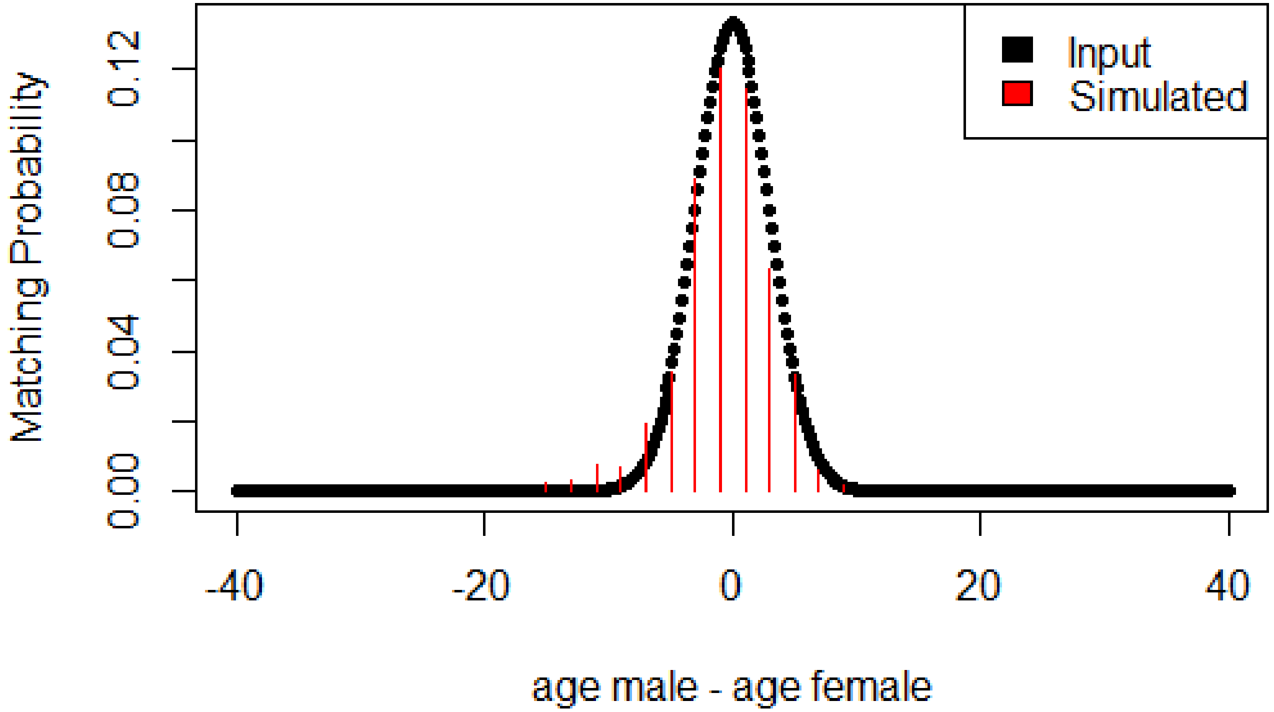
```
Initialization ...
[1] "Starting at:  2024-01-02 19:22:34"
[1] "Ending at:  2024-01-02 19:37:07"
Simulation is running ...
Year:  2014
Year:  2015
Year:  2016
Year:  2017
Year:  2018
     ...
Year:  2061
Year:  2062
Year:  2063
Year:  2064
Simulation has finished.
Number of unsuccessful partner search events:  0 of 24555 ( 0 % )
```

# Output: Virtual Population with Linked Individuals

| ID | birthDate | initState | From | To | transitionTime | transitionAge | motherID | fatherID | partnerID |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 19970920 | m/PH/0 | m/PH/0 | m/A/0 | 20140826 | 16.93 | NA | NA | NA |
| 1 | 19970920 | m/PH/0 | m/A/0 | m/PH/0 | 20311116 | 34.16 | NA | NA | NA |
| 1 | 19970920 | m/PH/0 | m/PH/0 | m/PA/0 | 20360824 | 38.93 | NA | NA | 26471 |
| 1 | 19970920 | m/PH/0 | m/PA/0 | dead | 20621215 | 65.24 | NA | NA | 26471 |
| 1 | 19970920 | m/PH/0 | m/PH/0 | m/A/0 | 20330205 | 35.38 | NA | NA | NA |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 100993 | 20140923 | f/PH/0 | f/PH/0 | f/A/0 | 20301215 | 16.23 | 96281 | 62592 | NA |
| 100993 | 20140923 | f/PH/0 | f/A/0 | f/A/1 | 20350424 | 20.59 | 96281 | 62592 | NA |
| 100993 | 20140923 | f/PH/0 | f/A/1 | f/PA/1 | 20360206 | 21.37 | 96281 | 62592 | 37891 |
| 100993 | 20140923 | f/PH/0 | f/PA/1 | f/PA/1 | 20430622 | 28.75 | 96281 | 62592 | 37891 |
| 100993 | 20140923 | f/PH/0 | f/PA/1 | f/PA/1 | 20500214 | 35.39 | 96281 | 62592 | 37891 |

# Output: Age Difference Distribution of Mates



# Next Steps

- Extend for more flexible mating rules (not only along age differences)
- Fully include into MicSim Package and upload to CRAN
- Conducting meaningful applications (e.g. upcoming VW project on inheritance of home ownership)

**Caution**

- Finding proper matches requires a well filled mating pool
- This requires a surplus of mate candidates (to ensure choice from potential mates with proper attributes)
- In female centered model: higher proportion of males in population necessary (stratification / *oversampling*)

**Software and teaching material incl. application examples**

R package MicSim (Version 2.0.0, last update 2023): Performing Continuous-Time Microsimulation.
http://cran.rproject.org/web/packages/MicSim/index.html

MicSim Package: Vignette

GitHub Repositories: https://github.com/bieneSchwarze

Especially:
- MicSim
- MicSimCourse
- MicSimLink

**Kontakt:**
szinn@diw

# Life-Course Model: Continuous Time Markov Multi State Model

***Stochastic process*** that at any point in time occupies one out of a set of discrete states

***Key quantities***: transition rates $\lambda_{sj,sk}(t)$ of Markovian process $Z(t)$ defined by $(J_n, T_n)_{n \in N_o}$

$$\lambda_{sj,sk}(t) = \lim_{h \downarrow 0} P[J_{n+1} = s_k, T_{n+1} \in (t, t+h] \mid J_n = s_j, T_{n+1} \geq t]$$

# Life-Course Model: Continuous Time Markov Multi State Model

***Stochastic process*** that at any point in time occupies one out of a set of discrete states

***Key quantities***: transition rates $\lambda_{sj,sk}(t)$ of Markovian process $Z(t)$ defined by $(J_n, T_n)_{n \in N_o}$

$$\lambda_{sj,sk}(t) = \lim_{h \downarrow 0} P[J_{n+1} = s_k, T_{n+1} \in (t, t+h] \mid J_n = s_j, T_{n+1} \geq t]$$

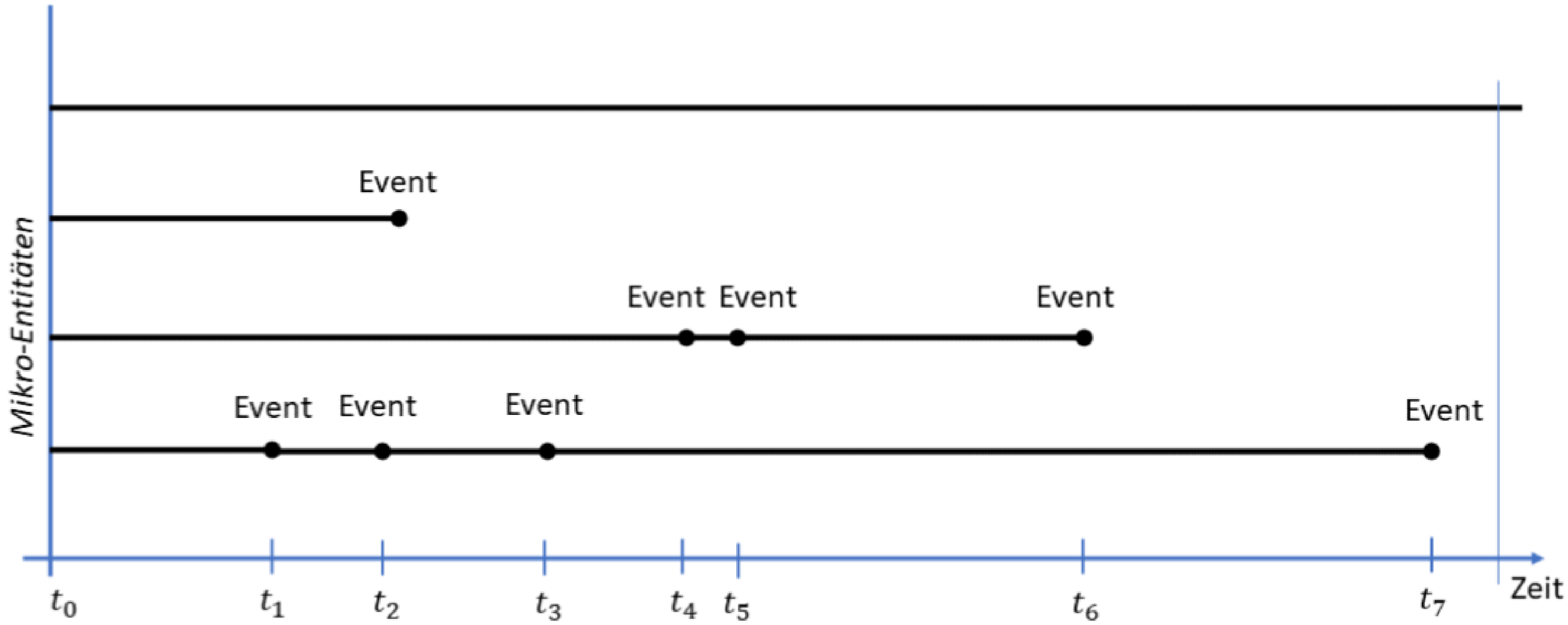
Allow deriving ***distribution function*** $F(w_{sj}, t) = 1 - S(w_{sj}, t)$ ***of waiting time*** to next event

The probability that an individual is still in state $s_j$ at time $t$ after waiting time $w_{sj}$ depends on all $K$ ***competing risks***

$$S(w_{sj}, t) = \prod_{k=1, k \neq j}^{K} \exp\big(-\Lambda(w_{sj,sk}, t)\big), \quad \Lambda\big(w_{sj,sk}, t\big) = \int_t^{w_{sj,sk}} \lambda_{sj,sk}(v) dv$$

with $w_{sj,sk}$ is the waiting time in $s_j$ after moving to $s_k$

# Handling of Time: Diskrete Events



# Output: Number of Events

```
# ------------------------------------------------------------------------------
# Have a look at the outcome
# ------------------------------------------------------------------------------
# Convert to Long format
popLong <- convertToLongFormat(pop)
table(popLong$OD)
```

```
 0->1   1->1  A->PA  A->PH   cens   dead  PA->A PA->PH   PH->A PH->PA
29831  34943  25933  71753 103009  61766  15451  22448 175786  23177
```