

State-transition models in R

Decision Modeling for Public Health Workshop

December 7, 2022

© Copyright 2017, THE HOSPITAL FOR SICK CHILDREN AND THE COLLABORATING INSTITUTIONS.

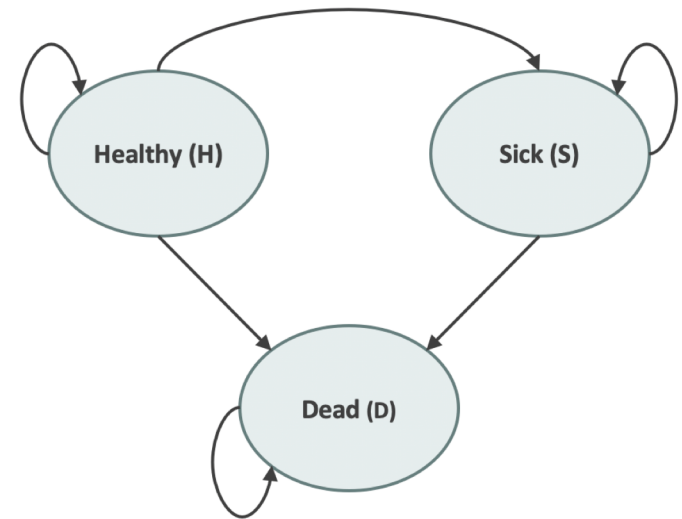
All rights reserved in Canada, the United States and worldwide. Copyright, trademarks, trade names and any and all associated intellectual property are exclusively owned by THE HOSPITAL FOR Sick CHILDREN and the collaborating institutions. These materials may be used, reproduced, modified, distributed and adapted with proper attribution.

The aim of this session:

- to give a short recap of the theory of state-transition models
- before we dive into coding in R

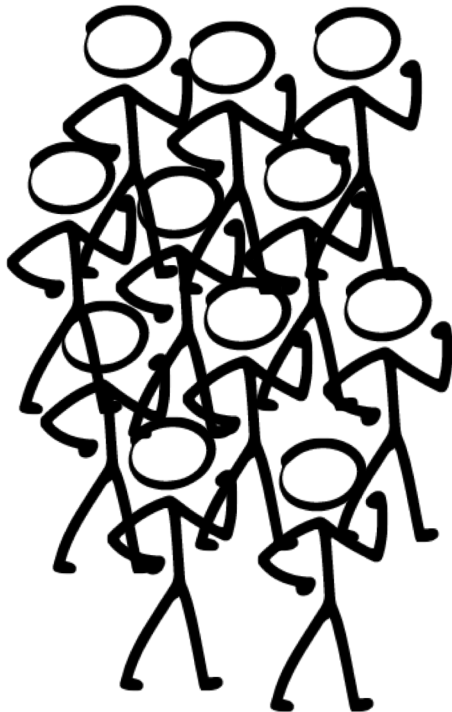
State-Transition Models

- Describes disease progression using a set collectively exhaustive and mutually exclusive health states
- Transitions allowed between health states with some probability
- Transitions occur in discrete time cycles (months, years, etc.)



Cohort and individual based state-transition models

Cohort



Individuals



Cohort State-Transition Models



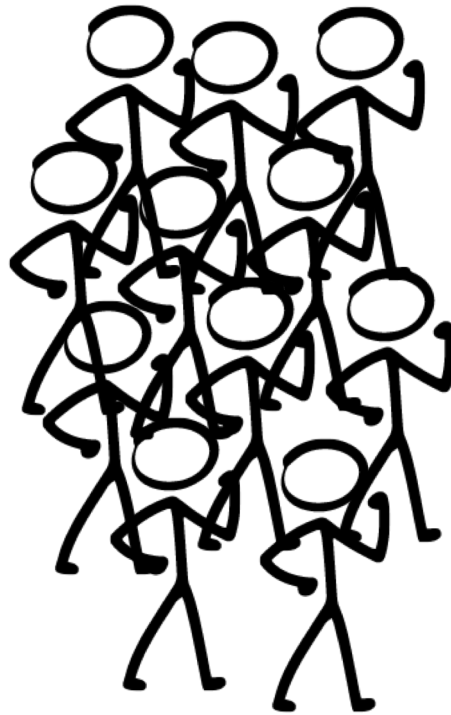
- Dynamic model that reflects disease progression and other events in a **cohort**
- Advantages:
 - Represents dynamic processes
 - Still relatively simple and so computationally efficient
- Best for dynamic processes that
 - Can be represented with a reasonable number of states
 - Don't dependent too much on individual heterogeneity (e.g., homogeneous cohort is a good approximation)
 - Minimal dependence on clinical history or time-in-state
- Deterministic model that generate mean outcomes

Individual based state-transition/ “Microsimulation”



- Stochastic dynamic model that simulates **individuals**, usually as a closed population
- Advantages:
 - Represents ***stochastic*** dynamic processes
 - Most flexible model type
 - Can capture ***complex dependencies*** on individual features, clinical history, time-since-event
- Best when a state-transition model is not sufficient
- Disadvantages
 - Computationally intensive, but can leverage parallel computing
 - Data intensive

Cohort state-transition model implementation in R



Transition Matrix Calculations

- Summarize transition probabilities as a matrix

		To				
		Healthy	Sick	Dead		
From:	Healthy	0.75	0.20	0.05	=	A
	Sick	0	0.85	0.15		
	Dead	0	0	1.0		

- Cohort distribution at next time step calculated through matrix multiplication

$$\begin{bmatrix} \text{Healthy} & x_{t+1} & \text{Sick} & \text{Dead} \end{bmatrix} = \begin{bmatrix} \text{Healthy} & x_t & \text{Sick} & \text{Dead} \end{bmatrix} \begin{bmatrix} A \end{bmatrix}$$

Transition Matrix Calculations

- Summarize transition probabilities as a matrix

		To				
		Healthy	Sick	Dead		
From:	Healthy	0.75	0.20	0.05	=	A
	Sick	0	0.85	0.15		
	Dead	0	0	1.0		

- Cohort distribution at next time step calculated through matrix multiplication

$$\begin{bmatrix} \text{Healthy} & x_{t+1} & \text{Sick} & \text{Dead} \end{bmatrix} = \begin{bmatrix} \text{Healthy} & x_t & \text{Sick} & \text{Dead} \end{bmatrix} \begin{matrix} A \\ \begin{bmatrix} 0.75 & 0.20 & 0.05 \\ 0 & 0.85 & 0.15 \\ 0 & 0 & 1.0 \end{bmatrix} \end{matrix}$$

Transition Matrix Calculations

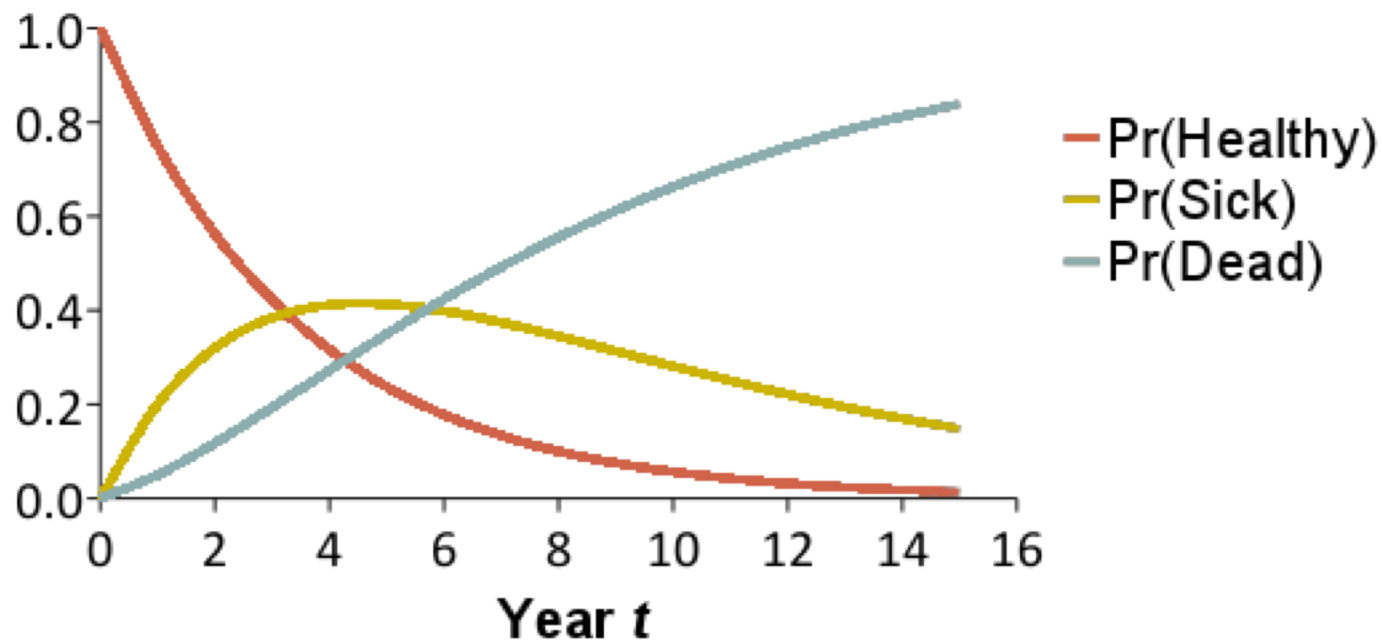
- Summarize transition probabilities as a matrix

		To			
		Healthy	Sick	Dead	
From:	Healthy	0.75	0.20	0.05	= A
	Sick	0	0.85	0.15	
	Dead	0	0	1.0	

- Cohort distribution at next time step calculated through matrix multiplication

$$\begin{matrix} & x_1 \\ \begin{bmatrix} 0.75 & 0.20 & 0.05 \end{bmatrix} & = & \begin{bmatrix} \textcircled{1.0} & 0.0 & 0.0 \end{bmatrix} & \begin{matrix} A \\ \begin{bmatrix} 0.75 & 0.20 & 0.05 \\ 0 & 0.85 & 0.15 \\ 0 & 0 & 1.0 \end{bmatrix} \end{matrix}
 \end{matrix}$$

Results in a Markov trace



Calculate the model Outcomes

- At each cycle, multiply state-specific outcomes (utilities & costs) by proportion of cohort in each state
- In CEA, outcomes are discounted to reflect time preferences
 - Prefer benefits now, costs later
 - Typical discount rate of 3% per year in US

Individuals



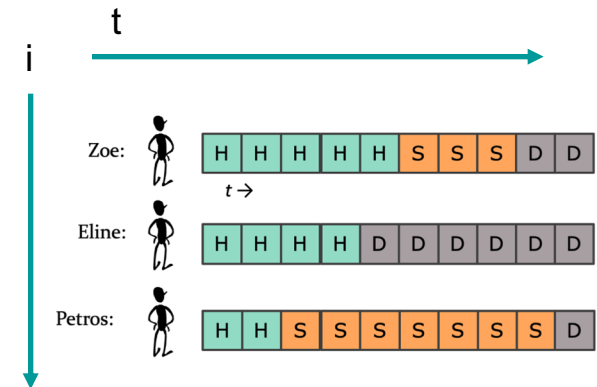
Implementation

- Simulating one individual at a time is an intuitive approach...

```
for(i in 1:n_i) { # open individual loop
  for (t in 1:n_t) { # open time loop

    # simulation code here #

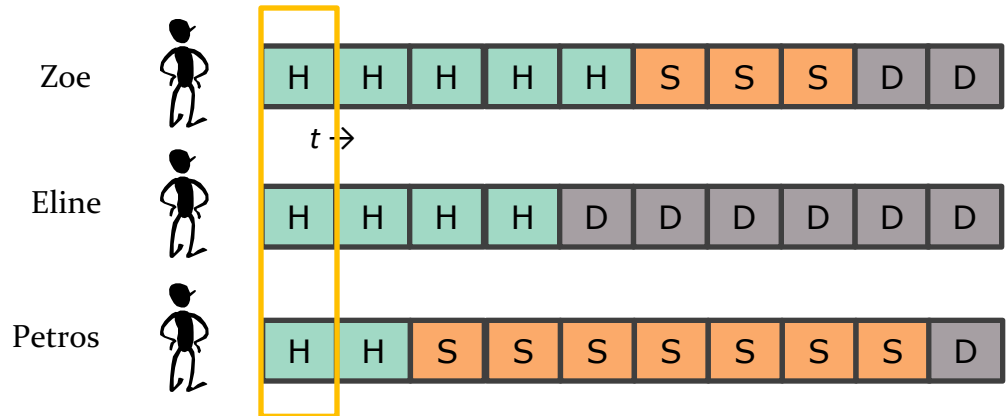
  } # close time loop
} # close individual loop
```



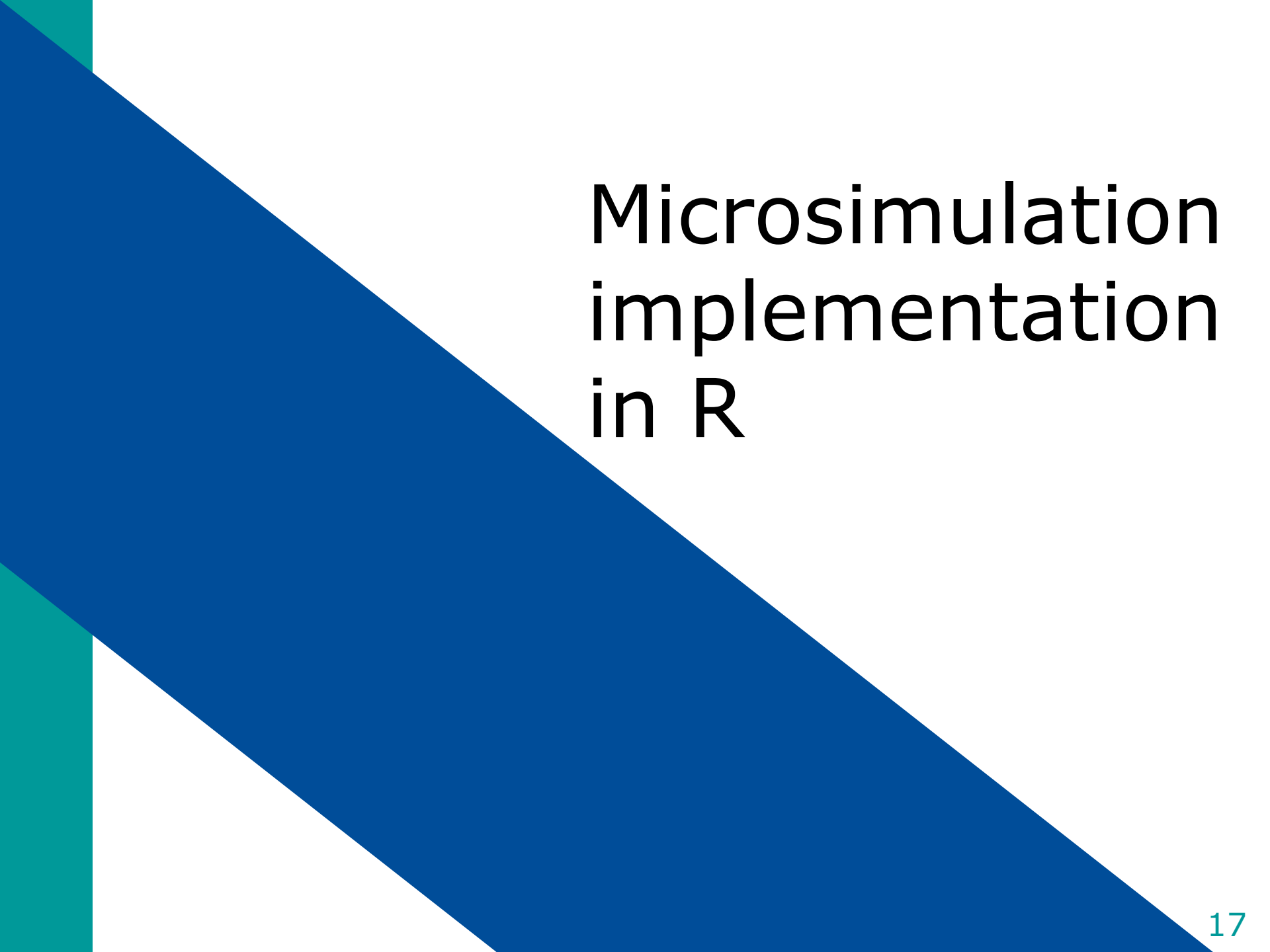
.... however, inefficient and can make a microsimulation computationally intensive

Individual characteristics

$t = 1$



	age	sex	height	Country of birth	time Sick	p_SD
Zoe	35	Female	1.55	Taiwan	0	--
Eline	28	Female	1.68	NL	0	--
Petros	36	Male	1.89	Greece	0	--



Microsimulation implementation in R

Functions

```
calculateMean <- function (x){  
  mean <- sum(x)/length(x)  
}
```

Structure of our code

Specify all the input parameters

- transition probabilities, cycle length etc