

# 9. Markov models

**Gianluca Baio**

Department of Statistical Science | University College London

- ✉ [g.baio@ucl.ac.uk](mailto:g.baio@ucl.ac.uk)
- 🌐 <https://gianluca.statistica.it/>
- 🌐 <https://egon.stats.ucl.ac.uk/research/statistics-health-economics/>
- 🔗 <https://github.com/giabaio>
- 🔗 <https://github.com/StatisticsHealthEconomics>
- 🐦 [@gianlubaio](https://twitter.com/gianlubaio)

STAT0019 - Bayesian Methods in Health Economics, UCL

- Assess **long-term** cost-effectiveness based only on **short-term** data
- State-transition (usually *Markov*) models for clinical histories
- Commonly implemented in Excel, or specialized software (eg TreeAge)
- Bayesian framework lets you **simultaneously** perform
  - parameter estimation (from short-term data, eg meta-analysis), and
  - probabilistic sensitivity analysis (long-term costs and benefits)
  - uncertainty about parameters fully included
- Example: 3-state cancer Markov model
- Markov models & survival analysis

## References

- *Bayesian Methods in Health Economics*, chapter 5.5  [Library](#)  [Book website \(CRC\)](#)  [Book website](#)  [Code](#)
- *Decision Modelling for Health Economic Evaluation*  [Library](#)  [Book website](#)
- *Evidence Synthesis for Decision Making in Healthcare*  [Library](#)  [Book website](#)
- *Bayesian Cost-Effectiveness Analysis with the R package BCEA*  [Book website \(Springer\)](#)  [Book website](#)

- Assume a set  $\mathcal{S}$  made of  $S$  "clinically relevant" states
  - Exhaustive and mutually exclusive
- The structure (links among nodes) describes the dynamics of disease history
  - Arrows connecting two states encode the assumption that a transition from the one where the arrow originates to the one reached by it is possible
  - Absence of an arrow between two states implies that the transition from one to the other is not allowed by our model

- Assume a set  $\mathcal{S}$  made of  $S$  "clinically relevant" states
  - Exhaustive and mutually exclusive
- The structure (links among nodes) describes the dynamics of disease history
  - Arrows connecting two states encode the assumption that a transition from the one where the arrow originates to the one reached by it is possible
  - Absence of an arrow between two states implies that the transition from one to the other is not allowed by our model
- From one period to the next, subjects can move among the states according to the rules specified by the arrows
- Movements occur according to suitable transition probabilities

$$\boldsymbol{\pi}_j = \boldsymbol{\pi}_{j-1} \boldsymbol{\Lambda}_j$$

where

- $\boldsymbol{\pi}_j = (\pi_{1j}, \dots, \pi_{Sj})$  is the vector of probabilities for each state at time  $j$
- $\boldsymbol{\Lambda}_j = [\Lambda_{j;s',s}]$  is a transition matrix describing the probability of moving from state  $s$  to state  $s'$  at time  $j$

- Assume a set  $\mathcal{S}$  made of  $S$  "clinically relevant" states
  - Exhaustive and mutually exclusive
- The structure (links among nodes) describes the dynamics of disease history
  - Arrows connecting two states encode the assumption that a transition from the one where the arrow originates to the one reached by it is possible
  - Absence of an arrow between two states implies that the transition from one to the other is not allowed by our model
- From one period to the next, subjects can move among the states according to the rules specified by the arrows
- Movements occur according to suitable transition probabilities

$$\boldsymbol{\pi}_j = \boldsymbol{\pi}_{j-1} \boldsymbol{\Lambda}_j$$

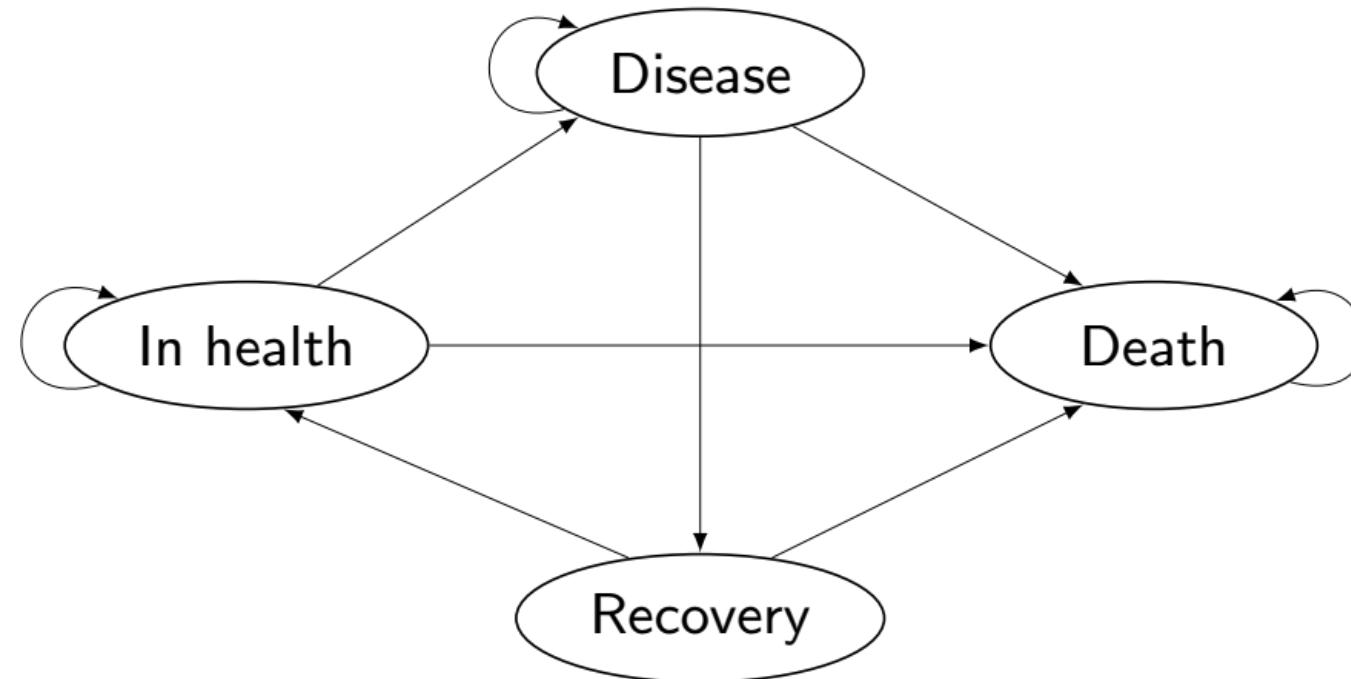
where

- $\boldsymbol{\pi}_j = (\pi_{1j}, \dots, \pi_{Sj})$  is the vector of probabilities for each state at time  $j$
- $\boldsymbol{\Lambda}_j = [\Lambda_{j;s',s}]$  is a transition matrix describing the probability of moving from state  $s$  to state  $s'$  at time  $j$
- **NB** the matrix algebra simply computes for each state  $s$

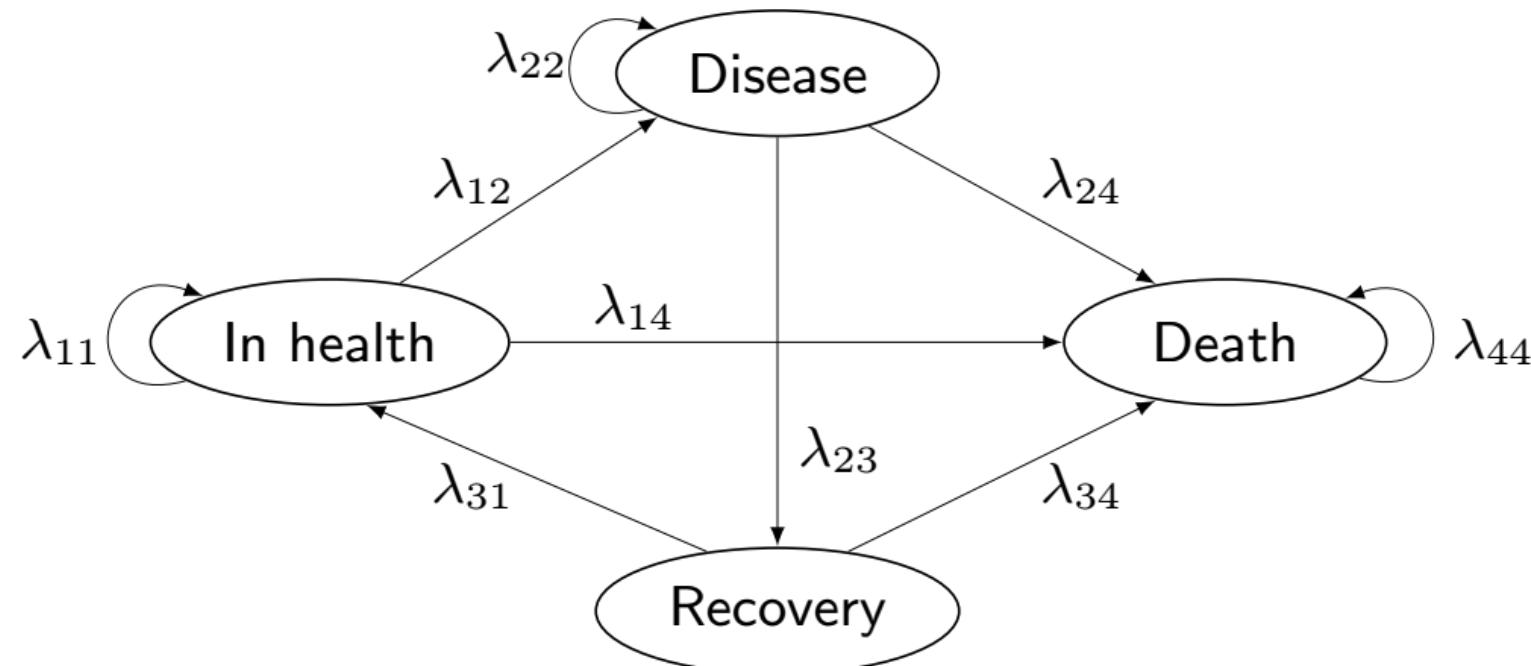
$$\Pr(\text{Being in state } s \text{ at time } j) = \sum_{s' \in \mathcal{S}} \Pr(\text{Being in state } s' \text{ at time } j-1) \times \Pr(\text{Moving from state } s' \text{ to state } s)$$

## 1. Define a structure

("Natural history" of the disease)



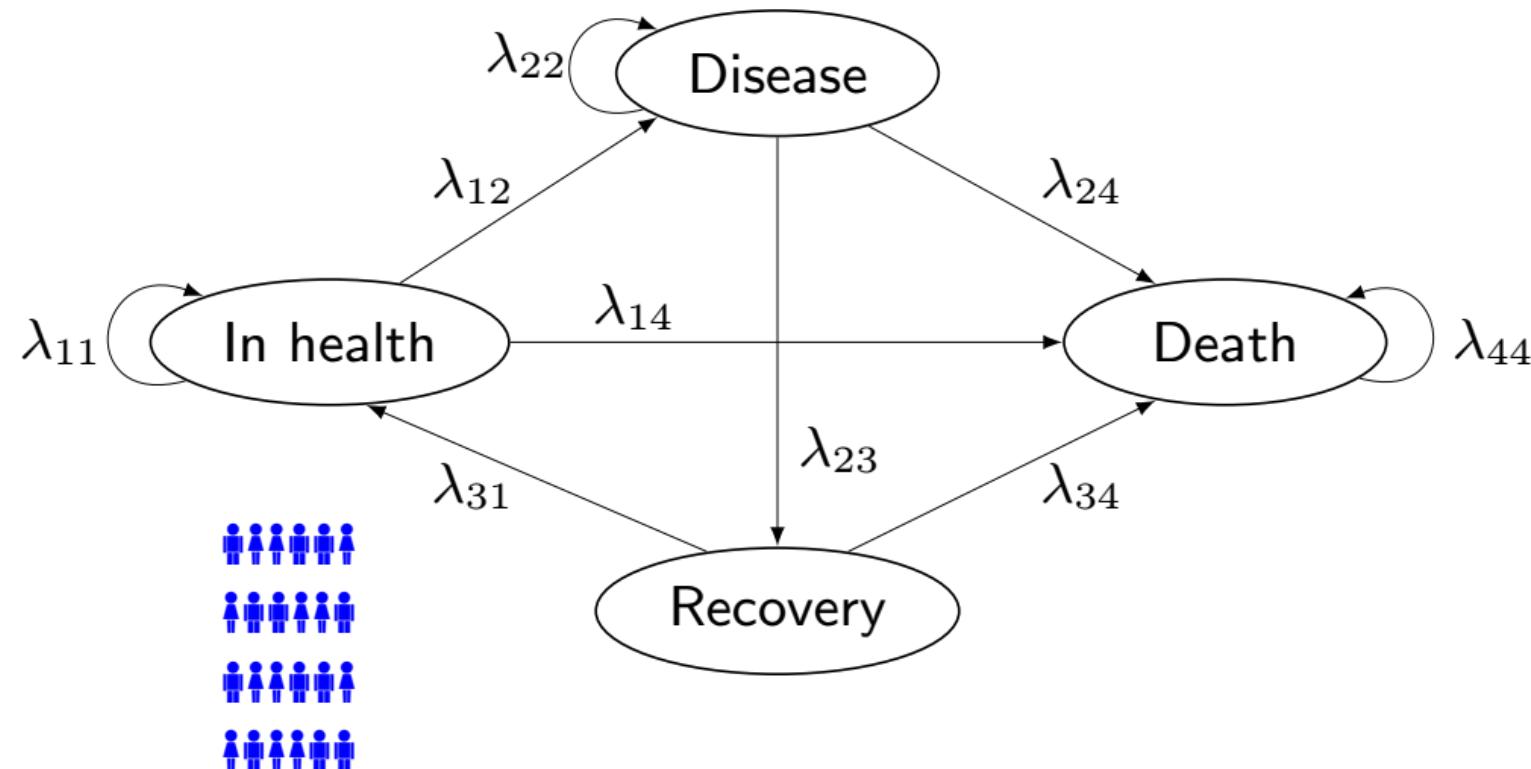
## 2. Estimate the transition probabilities



For instance:

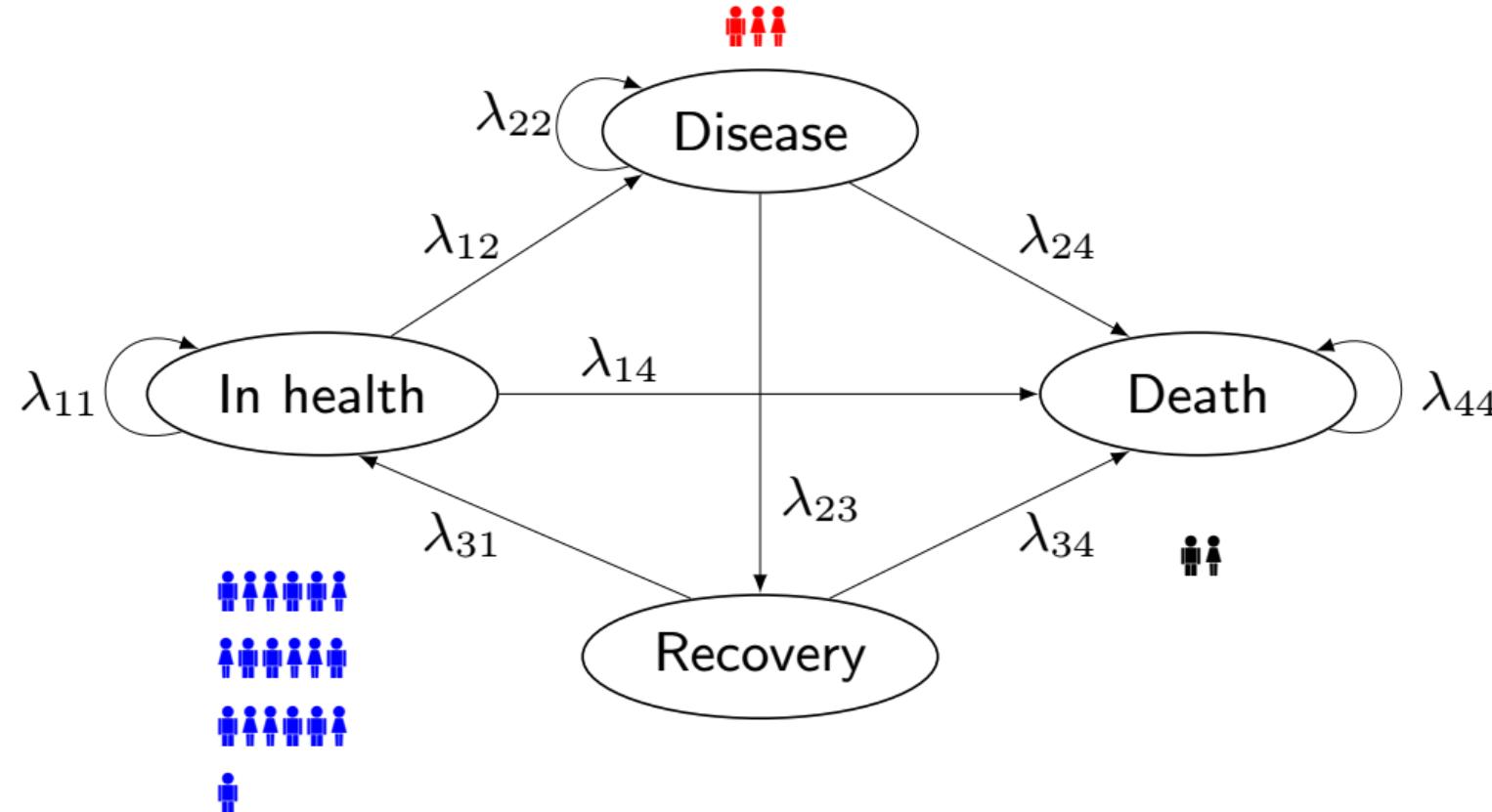
- $\lambda_{14}$  = general (healthy) population mortality  $\Rightarrow$  Relevant data: Life tables/official records, ...
- $\lambda_{24}$  = disease-specific mortality  $\Rightarrow$  Relevant data: Trial/observational studies, ...
- ...

3. Run the simulation:  $j = 0$



Distribute the "virtual cohort" across the  $S$  states (typically, everybody starts in the "healthy" state...)

3. Run the simulation:  $j = 1$



Start moving people around...

## Matrix algebra and "state occupancy"

- $m_{sj}$  is the **number** of people in state  $s$  at time  $j$
- $\lambda_{s'sj}$  is the probability of moving from state  $s'$  to state  $s$  between time  $j$  and  $j + 1$
- Thus:

$$m_{s'j+1} = m_{1j}\lambda_{1sj} + m_{2j}\lambda_{2sj} + \dots + m_{Sj}\lambda_{Ssj}$$

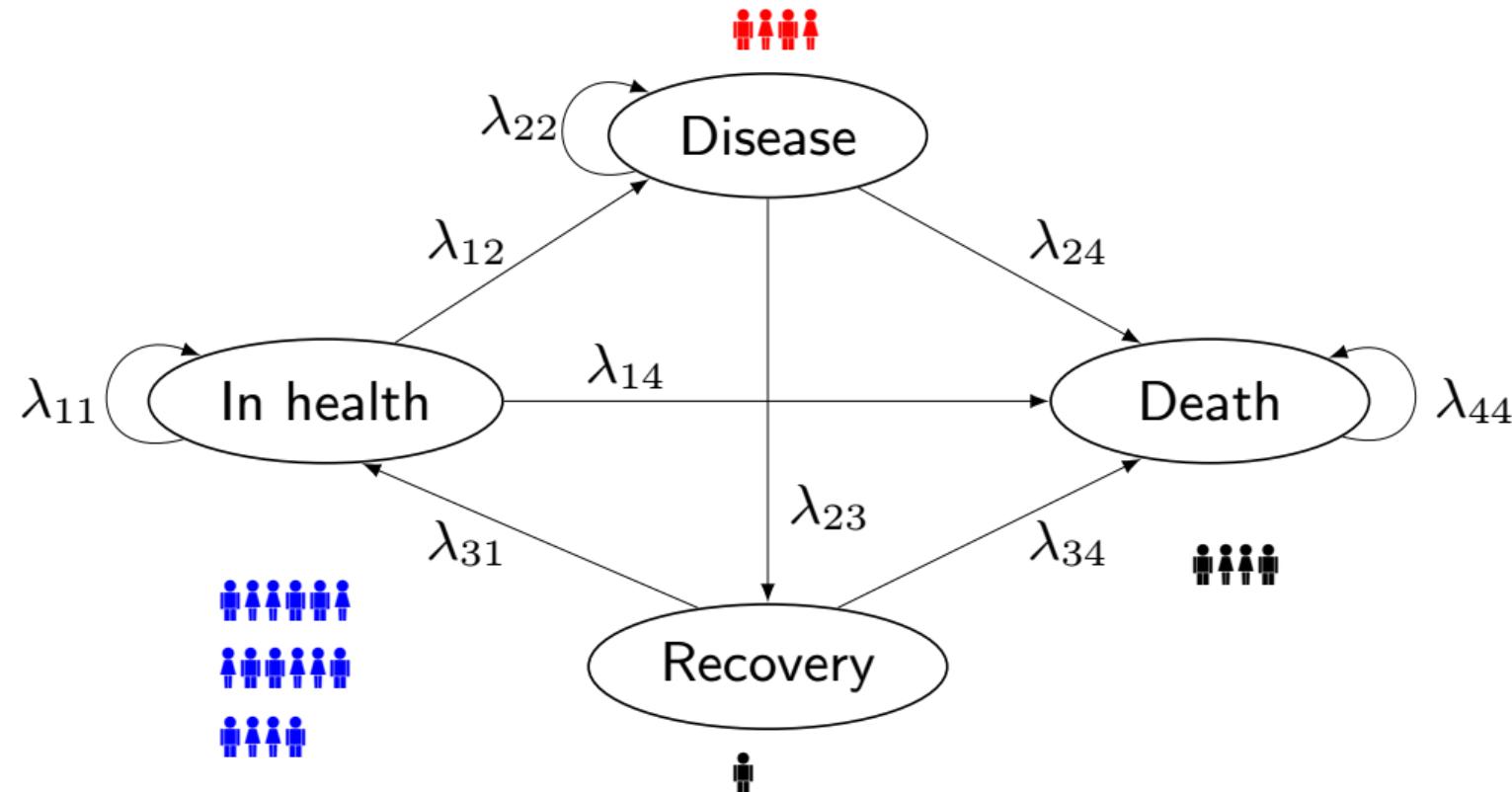
which we can write in matrix algebra as

$$(m_{1j+1}, \dots, m_{Sj+1}) = (m_{1j}, \dots, m_{Sj}) \begin{pmatrix} \lambda_{11j} & \dots & \lambda_{1Sj} \\ \vdots & \ddots & \vdots \\ \lambda_{S1j} & \dots & \lambda_{SSj} \end{pmatrix}$$

$$\mathbf{m}_{j+1} = \mathbf{m}_j \boldsymbol{\Lambda}_j$$

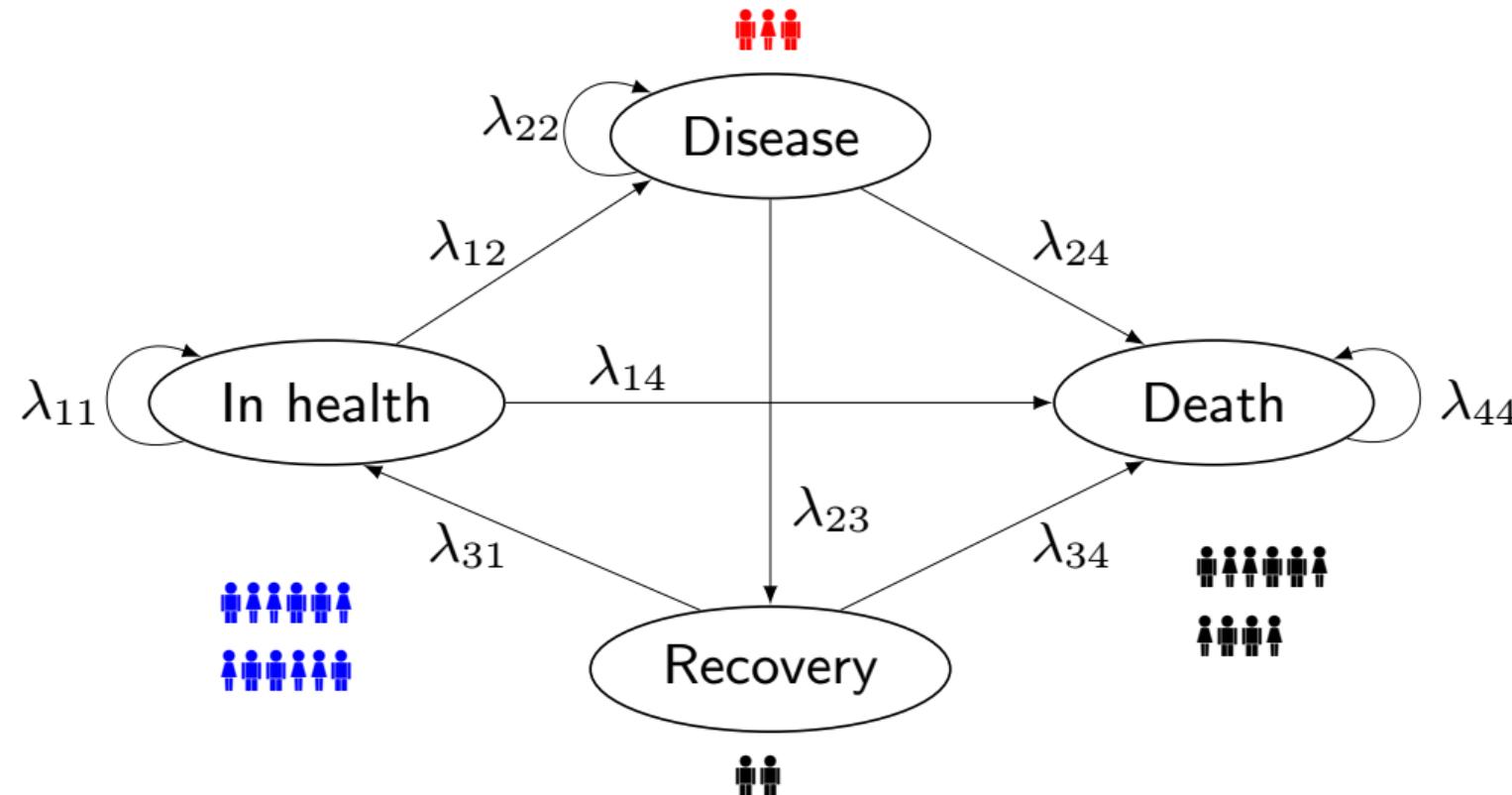
- **NB:** The transition matrix typically does depend on the time  $j$ , but sometimes we can relax this assumption

3. Run the simulation:  $j = 2$



Move people around according to the relationship  $\mathbf{m}_2 = \mathbf{m}_1 \Lambda_1$

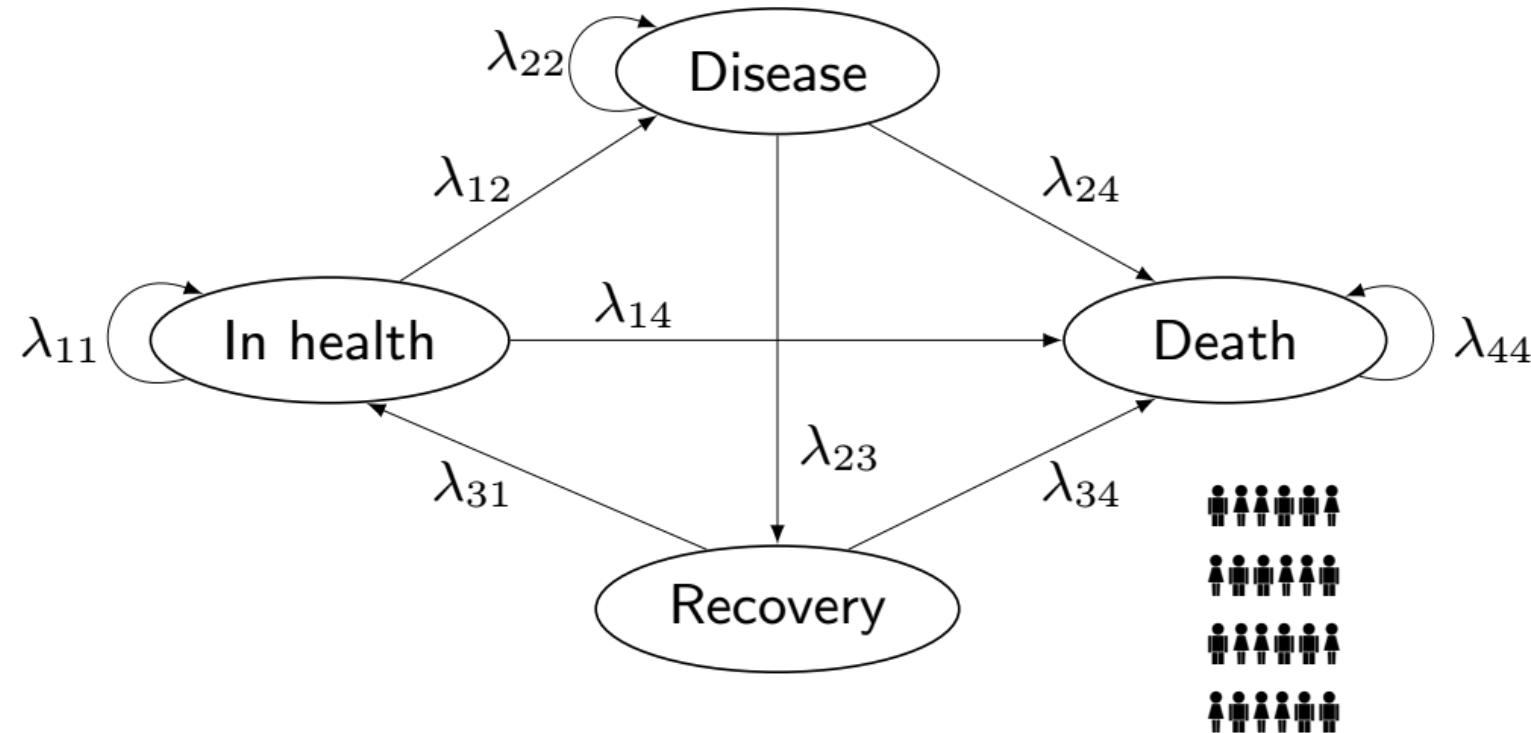
3. Run the simulation:  $j = 3$



Move people around according to the relationship  $\mathbf{m}_3 = \mathbf{m}_2 \boldsymbol{\Lambda}_2$

3. Run the simulation:  $j = J$

("*lifetime horizon*")



1 Assign **benefits** & **costs** to each state in the model and for each treatment  $t$  under study:  $(e_{ts}, c_{ts})$

- A measure of QoL (e.g. QALYs associated with being "perfectly healthy")
- A measure of cost (e.g. what does it cost the NHS for every person who has the disease?)

2 "Cohort simulation": estimate the proportion of individuals in each state at each time point (cycle)

- Usually need to do this for a long enough "virtual follow up" so that everybody reach the "absorbing state"
- That's a state from which you never move out (e.g. death)

3 For each treatment  $t$  under study, accumulate costs and benefits over time (slightly abusing the notation...)

$$\begin{aligned} e_t &= \sum_{j=0}^J \sum_{s=1}^S m_{tsj} e_{ts} && \text{and} & c_t &= \sum_{j=1}^J \sum_{s=1}^S m_{tsj} c_{ts} \\ &= \sum_{j=0}^J e_{tj} && & &= \sum_{j=0}^J c_{tj} \end{aligned}$$

NB: Costs and benefits can also be modelled to describe the uncertainty around their value

- **For example:** QALYs over one year  $\sim \text{Beta}(a, b)$ ; Costs  $\sim \text{Gamma}(\gamma, \rho)$ ; ...
- These may be informed by individual level data (see [Lecture 5](#)) or evidence synthesis (see [Lecture 7](#))

## Discounting

See BMHE, chapter 1.5; 5.4

- Costs and outcomes can occur at different times with respect to when the intervention is implemented
- **But:** society tends to value benefits that arrive closer to the present time more than those that will be achievable in the (possibly very distant) future
  - Example: Human Papilloma Virus (HPV) vaccination is available for boys and girls as young as 12, but benefits (protection from cervical cancer) only materialises when they are well in their 40s...

## Discounting

See BMHE, chapter 1.5; 5.4

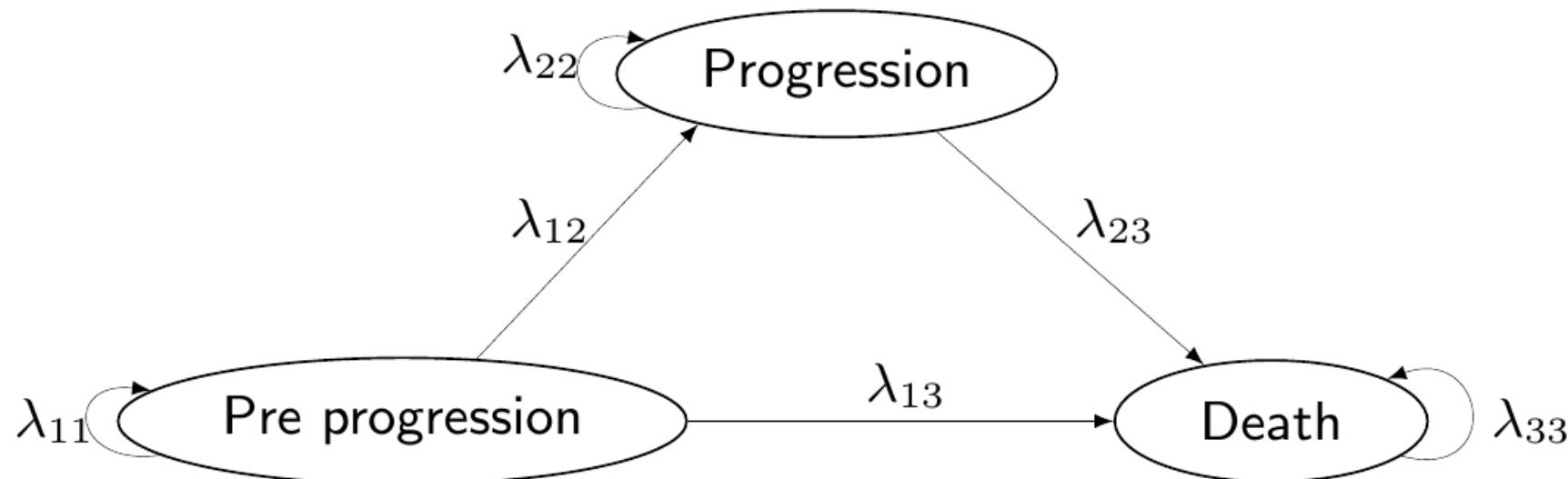
- Costs and outcomes can occur at different times with respect to when the intervention is implemented
- **But:** society tends to value benefits that arrive closer to the present time more than those that will be achievable in the (possibly very distant) future
  - Example: Human Papilloma Virus (HPV) vaccination is available for boys and girls as young as 12, but benefits (protection from cervical cancer) only materialises when they are well in their 40s...
- **Discounting** accounts for differential timing by reducing the value of costs and effects in the future
  - Particularly relevant for economic evaluation spanning over a time horizon  $> 1$  time unit (e.g. year)
  - Markov models tend to do that, so discounting is a key issue when using them!
- Define a **discount rate**  $d$  to ensure that costs sustained and benefits gained closer to now have more value
- Compute the **Present Value** of intervention  $t$  as

$$PV_t^e = \sum_{j=0}^J \frac{e_{tj}}{(1+d)^j} \quad \text{and} \quad PV_t^c = \sum_{j=0}^J \frac{c_{tj}}{(1+d)^j}$$

- NICE suggests using  $d = 3.5\%$  for costs and outcomes – but for some diseases can use different rates for costs and benefits...

## 3-state cancer Markov model

- This is one of the most popular structures developed as Markov models



- Patients typically enter the cohort in "pre-progression"
- Then can either die or "progress" to a worst condition
- From there they typically cannot revert to a better state

## 3-state cancer Markov model

Ideally, we can access individual level data (e.g. from a randomised trial), describing the "event history" for each patient

Patient	Treatment	Progression?	Death?	Progression time	Death time
1	1	1	0	31.99	32.00
2	1	1	0	30.55	30.60
...	...	...	...	...	...
10	1	1	1	0.17	0.46
11	1	1	1	1.27	1.57
...	...	...	...	...	...

This type of data would allow us to estimate directly all the relevant transition probabilities (with some extra work...)

- They contain information about the **complete history** of each individual and so we can use them to construct the "risk set" for each possible transition and then estimate the time-to-event

## Subset 1: Progression

- Risk set: all individuals who are at risk of making progression
  - The individuals in the risk set may or may not also die ("competing risks")
  - But we consider "Progression" as the **event** and "Death" or "No progression" as **censoring**

Code	Output	Examples (1)	Examples (2)
------	--------	--------------	--------------

```
> data %>% mutate(  
+   id=patid,                                     # patient ID  
+   from=1,                                       # starting state  
+   to=2,                                         # arriving state  
+   trans=1,                                       # transition code (1 = Pre -> Progression)  
+   Tstart=0,                                      # entry time  
+   Tstop=prog_t,                                    # exit time  
+   time=Tstop-Tstart,                             # time-to-event = Tstop-Tstart  
+   status=case_when(  
+     prog==1~1,                                     # censoring indicator:  
+     TRUE~0                                         # 1 if progressed; 0 otherwise  
+   ),  
+   treat=treat,                                     # treatment arm  
+ ) %>%  
+ # Selects only the relevant rows  
+ select(id,from,to,trans,Tstart,Tstop,time,prog,death,status,treat)
```

## Subset 1: Progression

- Risk set: all individuals who are at risk of making progression
  - The individuals in the risk set may or may not also die ("competing risks")
  - But we consider "Progression" as the **event** and "Death" or "No progression" as **censoring**

Code	Output	Examples (1)	Examples (2)
<hr/>			
# A tibble: 810 × 13			
	id from to trans Tstart Tstop time prog death status treat prog_t death_t		
1	<int> <dbl> <dbl> <dbl> <dbl> <dbl> <int> <int> <dbl> <int> <dbl>		
2	693 1 2 1 0 23.5 23.5 0 0 0 0 23.5		23.5
3	36 1 2 1 0 24.5 24.5 1 0 1 1 24.5		26
4	541 1 2 1 0 18.5 18.5 0 1 0 1 18.5		18.5
5	224 1 2 1 0 12.3 12.3 1 0 1 1 12.3		34
6	626 1 2 1 0 19.6 19.6 0 0 0 0 19.6		19.6
7	566 1 2 1 0 16.2 16.2 0 1 0 0 16.2		16.2
8	283 1 2 1 0 19.8 19.8 0 0 0 1 19.8		19.8
9	801 1 2 1 0 45 45 0 0 0 0 45		45
10	559 1 2 1 0 7.19 7.19 0 1 0 0 7.19		7.19
	429 1 2 1 0 31 31 0 0 0 1 31		31
# ... with 800 more rows			

## Subset 1: Progression

- Risk set: all individuals who are at risk of making progression
  - The individuals in the risk set may or may not also die ("competing risks")
  - But we consider "Progression" as the **event** and "Death" or "No progression" as **censoring**

Code	Output	Examples (1)	Examples (2)
------	--------	--------------	--------------

Patients who have progressed but not died, e.g.

```
# A tibble: 1 × 13
  id   from     to trans Tstart Tstop   time  prog death status treat prog_t death_t
<int> <dbl> <dbl> <dbl> <dbl> <dbl> <int> <int> <dbl> <int> <dbl> <dbl>
1    1      1     2     1      0    32.0  32.0     1     0     1     1    32.0    32
```

Patients who have died *at or after* progression, e.g.

```
# A tibble: 2 × 13
  id   from     to trans Tstart Tstop   time  prog death status treat prog_t death_t
<int> <dbl> <dbl> <dbl> <dbl> <dbl> <int> <int> <dbl> <int> <dbl> <dbl>
1    10     1     2     1      0  0.173  0.173     1     1     1     1  0.173  0.458
2   134     1     2     1      0  6.90   6.90      1     1     1     1  6.90   14.3
```

## Subset 1: Progression

- Risk set: all individuals who are at risk of making progression
  - The individuals in the risk set may or may not also die ("competing risks")
  - But we consider "Progression" as the **event** and "Death" or "No progression" as **censoring**

Code	Output	Examples (1)	Examples (2)
------	--------	--------------	--------------

Patients who have died before progression (NB: these are **censored** in this case!)

```
# A tibble: 2 × 13
  id from to trans Tstart Tstop time prog death status treat prog_t death_t
  <int> <dbl> <dbl> <dbl> <dbl> <dbl> <int> <int> <dbl> <int> <dbl> <dbl>
1 527     1     2     1     0   1.50  1.50     0     1     0     1   1.50  1.50
2 528     1     2     1     0   2.07  2.07     0     1     0     1   2.07  2.07
```

Patients who have been fully censored

```
# A tibble: 2 × 13
  id from to trans Tstart Tstop time prog death status treat prog_t death_t
  <int> <dbl> <dbl> <dbl> <dbl> <dbl> <int> <int> <dbl> <int> <dbl> <dbl>
1 249     1     2     1     0 0.0575 0.0575     0     0     0     1 0.0575 0.0575
2 250     1     2     1     0 3.80   3.80      0     0     0     1 3.80   3.80
```

## Subset 2: Death from pre-progression

- Risk set: all individuals who are at risk of dying
  - The individuals in the risk set may or may not also progress ("competing risks")
  - But we consider "Death" as the **event** and "Progression" or "No progression" as **censoring**

Code	Output	Examples (1)	Examples (2)
------	--------	--------------	--------------

```
> data %>% mutate(  
+   id=patid,                                     # patient ID  
+   from=1,                                       # starting state  
+   to=3,                                         # arriving state  
+   trans=2,                                       # transition code (2 = Pre -> Death)  
+   Tstart=0,                                      # entry time  
+   Tstop=prog_t,                                    # exit time  
+   time=Tstop-Tstart,                             # time-to-event = Tstop-Tstart  
+   status=case_when(  
+     (death==1 & prog_t==death_t)~1,               # censoring indicator:  
+     TRUE~0                                         # 1 if died at progression; 0 otherwise  
+   ),  
+   treat=treat,                                     # treatment arm  
+ ) %>%  
+ # Selects only the relevant rows  
+ select(id,from,to,trans,Tstart,Tstop,time,prog,death,status,treat)
```

## Subset 2: Death from pre-progression

- Risk set: all individuals who are at risk of dying
  - The individuals in the risk set may or may not also progress ("competing risks")
  - But we consider "Death" as the **event** and "Progression" or "No progression" as **censoring**

Code	Output	Examples (1)	Examples (2)										
<hr/>													
# A tibble: 810 × 13													
	id from to trans Tstart Tstop time prog death status treat prog_t death_t												
1	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<int>	<int>	<dbl>	<int>	<dbl>	<dbl>	
2	196	1	3	2	0	19.3	19.3	1	0	0	0	19.3	34.9
3	379	1	3	2	0	26.2	26.2	0	0	0	1	26.2	26.2
4	592	1	3	2	0	12.0	12.0	0	0	0	0	12.0	12.0
5	64	1	3	2	0	33.4	33.4	1	1	0	0	33.4	36.1
6	444	1	3	2	0	33	33	0	0	0	1	33	33
7	359	1	3	2	0	24	24	0	0	0	1	24	24
8	43	1	3	2	0	3.97	3.97	1	0	0	0	3.97	5.76
9	234	1	3	2	0	6.16	6.16	1	1	0	1	6.16	30.7
10	198	1	3	2	0	18.2	18.2	1	0	0	1	18.2	34
	340	1	3	2	0	22.5	22.5	0	0	0	1	22.5	22.5
# ... with 800 more rows													

## Subset 2: Death from pre-progression

- Risk set: all individuals who are at risk of dying
  - The individuals in the risk set may or may not also progress ("competing risks")
  - But we consider "Death" as the **event** and "Progression" or "No progression" as **censoring**

Code	Output	Examples (1)	Examples (2)
------	--------	--------------	--------------

Patients who have progressed, but not died, e.g.

```
# A tibble: 1 × 13
  id   from     to trans Tstart Tstop   time   prog death status treat prog_t death_t
<int> <dbl> <dbl> <dbl> <dbl> <dbl> <int> <int> <dbl> <int> <dbl> <dbl>
1    1      1     3     2      0    32.0  32.0     1     0     0     1    32.0    32
```

Patients who have died *at* or *after* progression (NB: these are **censored** in this case!), e.g.

```
# A tibble: 2 × 13
  id   from     to trans Tstart Tstop   time   prog death status treat prog_t death_t
<int> <dbl> <dbl> <dbl> <dbl> <dbl> <int> <int> <dbl> <int> <dbl> <dbl>
1    10     1     3     2      0  0.173  0.173     1     1     0     1  0.173  0.458
2   134     1     3     2      0  6.90   6.90      1     1     0     1  6.90   14.3
```

## Subset 2: Death from pre-progression

- Risk set: all individuals who are at risk of dying
  - The individuals in the risk set may or may not also progress ("competing risks")
  - But we consider "Death" as the **event** and "Progression" or "No progression" as **censoring**

Code	Output	Examples (1)	Examples (2)
------	--------	--------------	--------------

### Patients who have died before progression

```
# A tibble: 2 × 13
  id   from     to trans Tstart Tstop   time   prog death status treat  prog_t death_t
  <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <int> <int> <dbl> <int> <dbl> <dbl>
1 527     1     3     2     0   1.50  1.50     0     1     1     1   1.50  1.50
2 528     1     3     2     0   2.07  2.07     0     1     1     1   2.07  2.07
```

### Patients who have been fully censored

```
# A tibble: 2 × 13
  id   from     to trans Tstart Tstop   time   prog death status treat  prog_t death_t
  <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <int> <int> <dbl> <int> <dbl> <dbl>
1 249     1     3     2     0  0.0575  0.0575     0     0     0     1  0.0575  0.0575
2 250     1     3     2     0  3.80    3.80      0     0     0     1  3.80    3.80
```

## Subset 3: Death from progression

- Risk set: all individuals who have progressed and are at risk of dying
  - The individuals in the risk set have certainly progressed and may or may not also die
  - But we consider "Death" as the **event** and "No death" as **censoring**

Code	Output	Examples (1)
------	--------	--------------

```
> data %>% filter(prog==1) %>% mutate( # NB: Filter for patients who *have* progressed!  
+ id=patid, # patient ID  
+ from=2, # starting state  
+ to=3, # arriving state  
+ trans=3, # transition code (2 = Pre -> Death)  
+ Tstart=prog_t, # entry time  
+ Tstop=death_t, # exit time  
+ time=Tstop-Tstart, # time-to-event = Tstop-Tstart  
+ status=case_when(  
+   death==1~1, # censoring indicator:  
+   TRUE~0 # 1 if died; 0 otherwise  
+ ),  
+ treat=treat # treatment arm  
+ ) %>%  
+ # Selects only the relevant rows  
+ select(id,from,to,trans,Tstart,Tstop,time,prog,death,status,treat)
```

## Subset 3: Death from progression

- Risk set: all individuals who have progressed and are at risk of dying
  - The individuals in the risk set have certainly progressed and may or may not also die
  - But we consider "Death" as the **event** and "No death" as **censoring**

Code	Output	Examples (1)
<hr/>		
# A tibble: 248 × 13		
	id from to trans Tstart Tstop time prog death status treat prog_t death_t	
1	<int> <dbl> <dbl> <dbl> <dbl> <dbl> <int> <int> <dbl> <int> <dbl>	73 2 3 3 21.3 24.8 3.47 1 0 0 0 21.3 24.8
2	131 2 3 3 40.7 48 7.32 1 0 0 1 40.7 48	
3	245 2 3 3 12.6 45.1 32.5 1 0 0 0 12.6 45.1	
4	93 2 3 3 42.8 48 5.19 1 0 0 1 42.8 48	
5	214 2 3 3 15.4 34 18.6 1 0 0 1 15.4 34	
6	65 2 3 3 22.2 24.9 2.73 1 0 0 0 22.2 24.9	
7	151 2 3 3 5.93 14.5 8.55 1 1 1 1 5.93 14.5	
8	96 2 3 3 35.6 41 5.38 1 0 0 1 35.6 41	
9	163 2 3 3 10.2 19.3 9.08 1 1 1 1 10.2 19.3	
10	230 2 3 3 11.8 35 23.2 1 0 0 1 11.8 35	
# ... with 238 more rows		

## Subset 3: Death from progression

- Risk set: all individuals who have progressed and are at risk of dying
  - The individuals in the risk set have certainly progressed and may or may not also die
  - But we consider "Death" as the **event** and "No death" as **censoring**

Code	Output	Examples (1)
------	--------	--------------

Patients who have died after progression, e.g.

```
# A tibble: 1 × 13
  id   from     to trans Tstart Tstop   time  prog death status treat prog_t death_t
<int> <dbl> <dbl> <dbl> <dbl> <dbl> <int> <int> <dbl> <int> <dbl>    <dbl>
1    10      2     3     3  0.173  0.458  0.286     1     1     1     1  0.173  0.458
```

Patients who have not died (NB: these are **censored** in this case!), e.g.

```
# A tibble: 1 × 13
  id   from     to trans Tstart Tstop   time  prog death status treat prog_t death_t
<int> <dbl> <dbl> <dbl> <dbl> <dbl> <int> <int> <dbl> <int> <dbl>    <dbl>
1     1      2     3     3  32.0    32 0.00920     1     0     0     1  32.0    32
```

## Estimating the survival curves

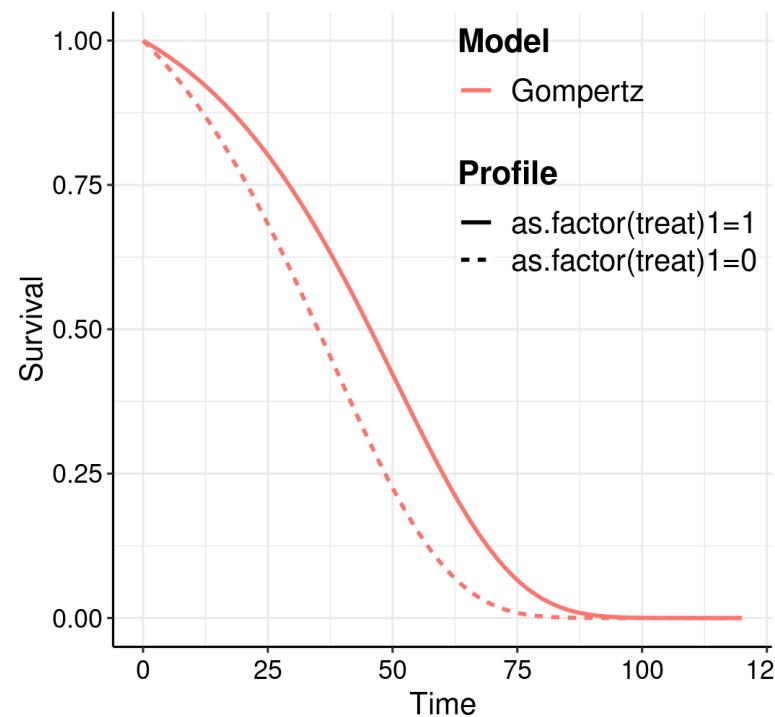
- We can use the models we saw in [Lecture 6](#) to estimate the survival curves for the 3 subsets
- For example, we can test various alternative and settle for a Gompertz distribution

```
> # Sets up informative priors on the Gompertz parameters to stabilise inference
> priors=list(gom=list(a_alpha=1.5,b_alpha=1.5))
>
> # Runs survival models on the specific subsets to obtain estimate of the various transition probabilities
> m_12=fit.models(Surv(time,status)~as.factor(treat),data=subdata1,distr="gom",method="hmc",priors=priors)
> m_13=fit.models(Surv(time,status)~as.factor(treat),data=subdata2,distr="gom",method="hmc",priors=priors)
> m_23=fit.models(Surv(time,status)~as.factor(treat),data=subdata3,distr="gom",method="hmc",priors=priors)
```

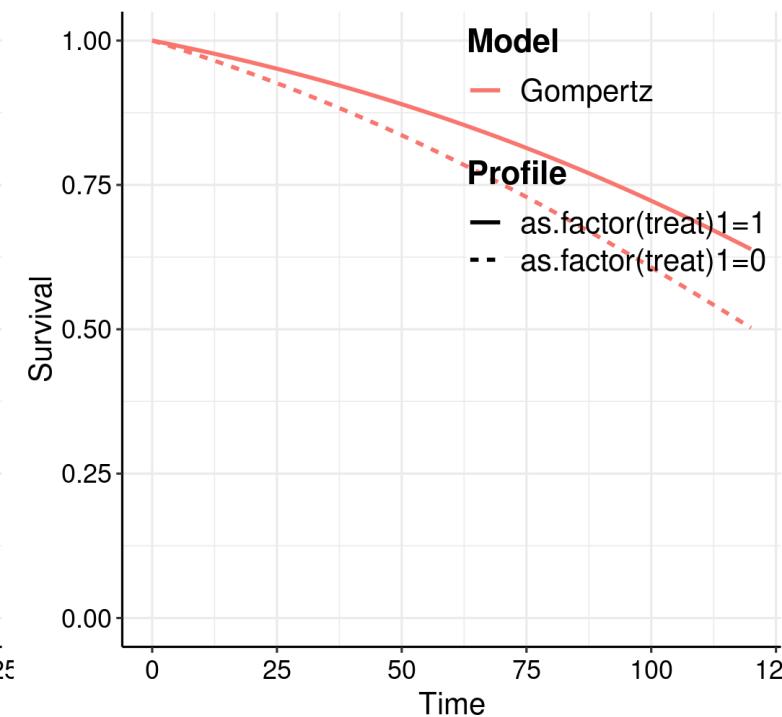
- Now can use the simulated values for the survival curves to *approximate* the transition probabilities
  - Technically, the models above estimate survival over continuous times, while the Markov model assumes discrete cycles
- Can use the `survHE` function `make.surv` to generate simulations from the survival curves from each model over any arbitrary extrapolation

# Survival curves and transition probabilities

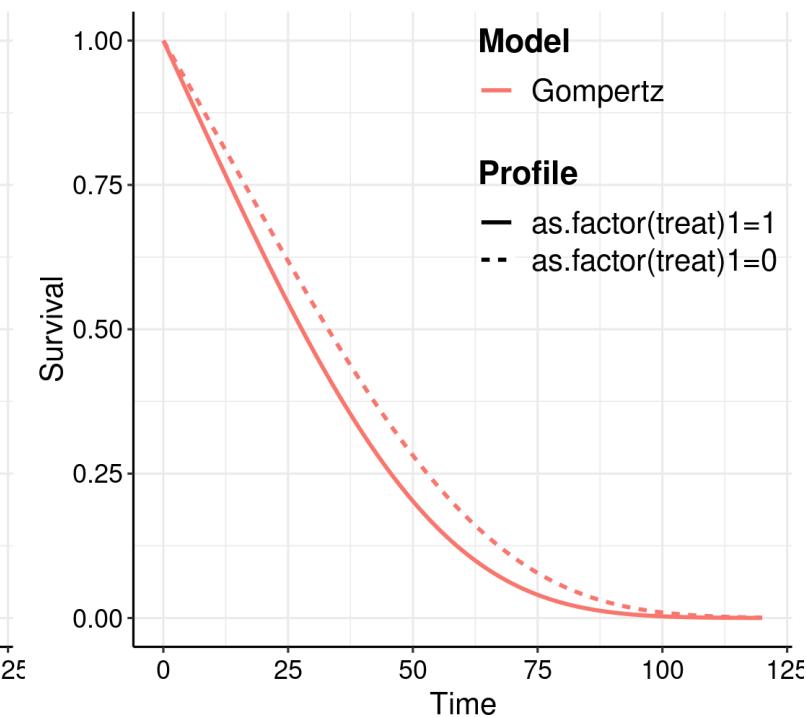
**Progression**



**Death from pre-progression**



**Death from progression**



In general, given a survival curve representing a suitable transition, we can compute

$$\lambda_{s'sj} \approx 1 - \frac{S_{j+k}}{S_j}$$

(intuitively, the transition probability can be read off as the reduction in the proportion of individuals who have *not* experienced the event between two consecutive times)

# Survival curves and transition probabilities

We can directly estimate

- $\lambda_{12j}$  from the survival curves derived with subset 1
- $\lambda_{13j}$  from the survival curves derived with subset 2
- $\lambda_{23j}$  from the survival curves derived with subset 3

Indirectly (using the fact that the transitions are mutually exclusive and exhaustive)

- $\lambda_{11j} = 1 - \lambda_{12j} - \lambda_{13j}$
- $\lambda_{22j} = 1 - \lambda_{23j}$

And because "Death" is an absorbing state, we can complete the transition matrix

$$\Lambda_j = \begin{pmatrix} \lambda_{11j} & \lambda_{12j} & \lambda_{13j} \\ 0 & \lambda_{22j} & \lambda_{23j} \\ 0 & 0 & 1 \end{pmatrix}$$

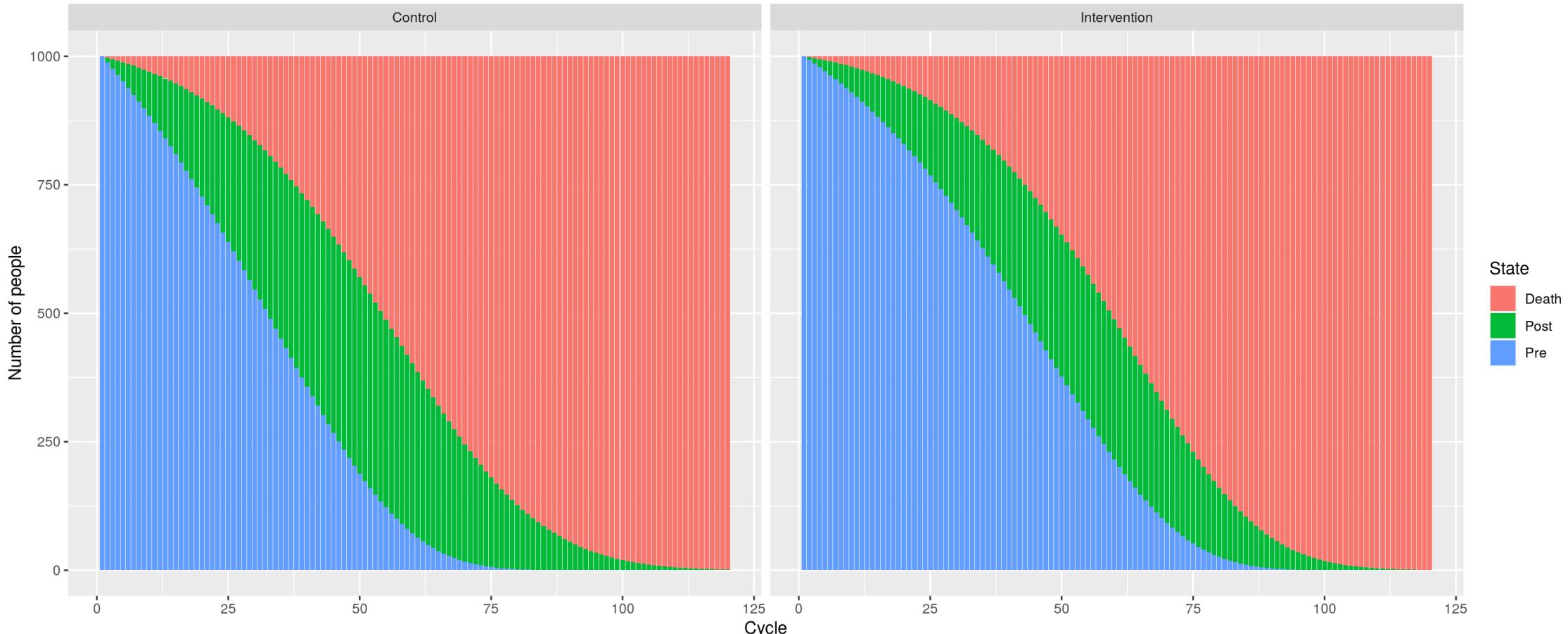
We can also replicate this calculation for  $n_{sim}$  times to propagate the uncertainty in the survival curves onto the whole Markov model

NB: May need more complex modelling (see for instance Williams et al, 2017)

# Running the Markov model

We can finally run the Markov model by initialising the state occupancy  $m_0$  and simply apply the matrix algebra to determine the number of people in each state at each time point.

Markov trace



## If we have the individual level data...

- In that case, we can recreate direct estimates of all the relevant transition probabilities
  - In that case, running the Markov model is basically just a problem of matrix algebra
  - Modellers tend to do this using a combination of statistical software (for the survival analysis) and spreadsheet (to compute the actual Markov model)
  - ... But of course this can all be done in a much more efficient way using a proper statistical software (e.g. R – see [Practical](#))

## If we have the individual level data...

- In that case, we can recreate direct estimates of all the relevant transition probabilities
  - In that case, running the Markov model is basically just a problem of matrix algebra
  - Modellers tend to do this using a combination of statistical software (for the survival analysis) and spreadsheet (to compute the actual Markov model)
  - ... But of course this can all be done in a much more efficient way using a proper statistical software (e.g. R – see [Practical](#))

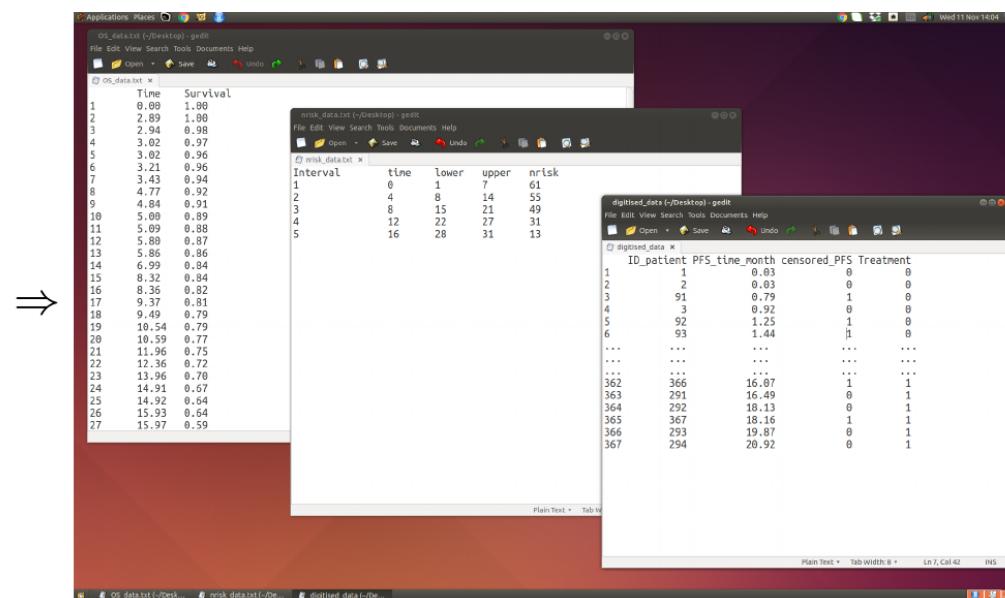
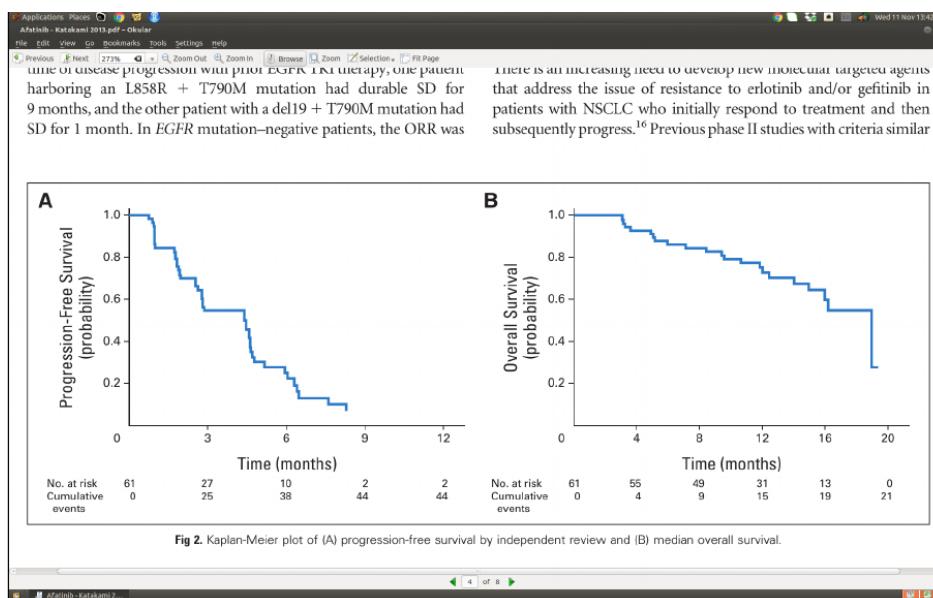
## What if we don't get the trial data?...

- Unfortunately, very often we are **not** in a position of using individual level data
  - Either because we have trial data for "our" study, but not for the comparators
  - Or because we do not have access to the ILD for any of the relevant treatments
- In these cases, we need to resort to **sub-optimal** methods of analysis
  - Specifically, we can use published summaries to obtain "pseudo-data" based on published Kaplan-Meier estimates
  - This practice is becoming very popular and is often termed as "modelling based on **digitised** data"

- Use specialised software to extract data values from published graphs
  - Example: DigitizeIT

# Digitised data

- Use specialised software to extract data values from published graphs
  - Example:  **DigitizeIT**
- Point & click on the curves from published papers
- Save suitable text files that can be fed to appropriate R scripts and algorithms to reconstruct the underlying Kaplan Meier curves  **Guyot et al, 2012**



## What's wrong with that?...

The problem is that often published papers report data on PFS and OS, BUT:

- 1 The curves are reported **separately** and **independently**
  - Although the data are correlated, because it is the same individuals undergoing the various transitions, there is no way to recover this level of correlation from the digitised data
- 2 The reported curves do not allow to control/stratify for a large number of covariates
  - It is possible that KM curves are reported, say, for males and females, separately; but stratification for many other variables is rare
- 3 It is then impossible to subset the underlying data and estimate all the relevant transition probabilities!

## What's wrong with that?...

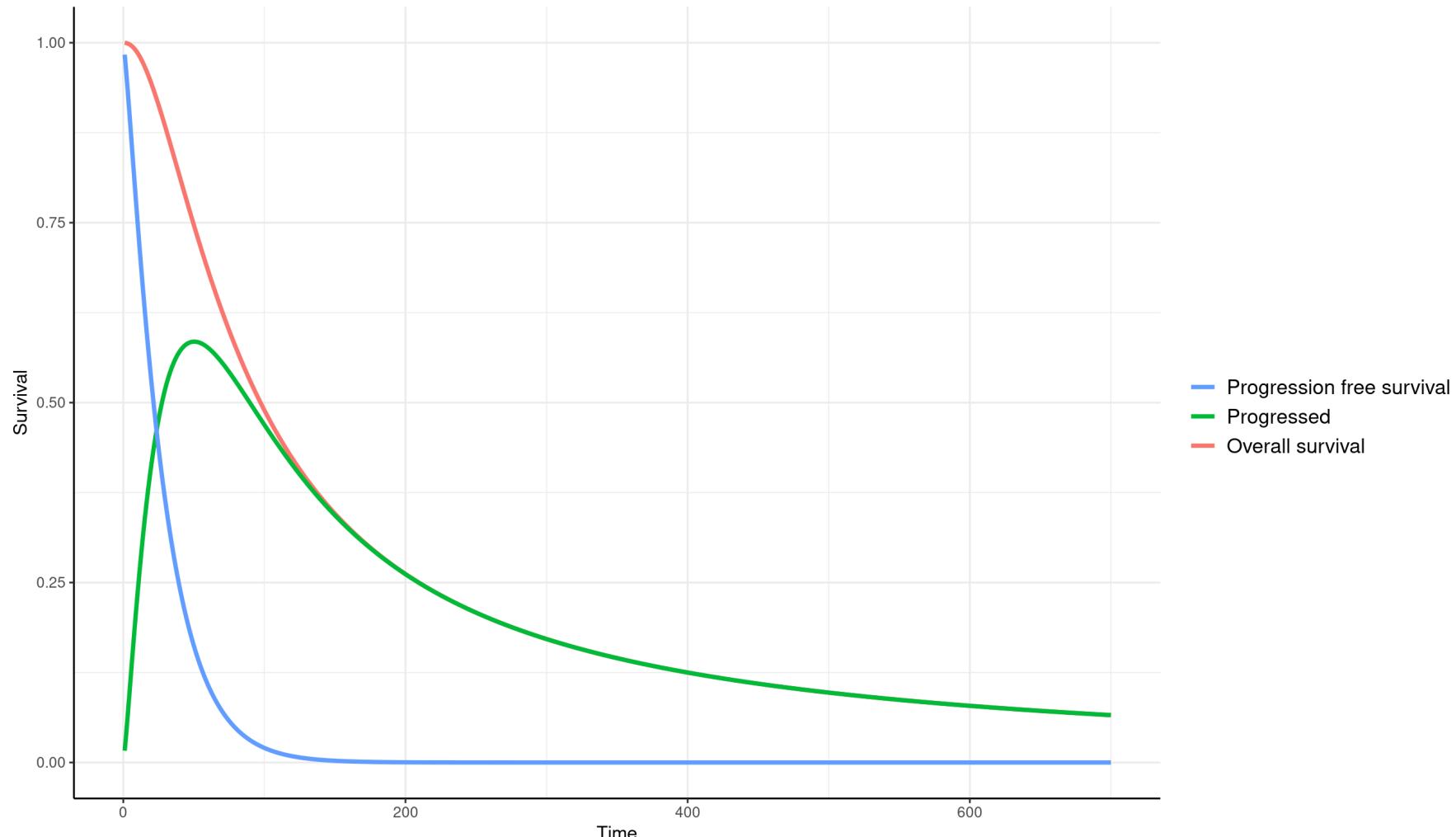
The problem is that often published papers report data on PFS and OS, BUT:

- 1 The curves are reported **separately** and **independently**
  - Although the data are correlated, because it is the same individuals undergoing the various transitions, there is no way to recover this level of correlation from the digitised data
- 2 The reported curves do not allow to control/stratify for a large number of covariates
  - It is possible that KM curves are reported, say, for males and females, separately; but stratification for many other variables is rare
- 3 It is then impossible to subset the underlying data and estimate all the relevant transition probabilities!

So?... ⇒ **Partitioned Survival Modelling**

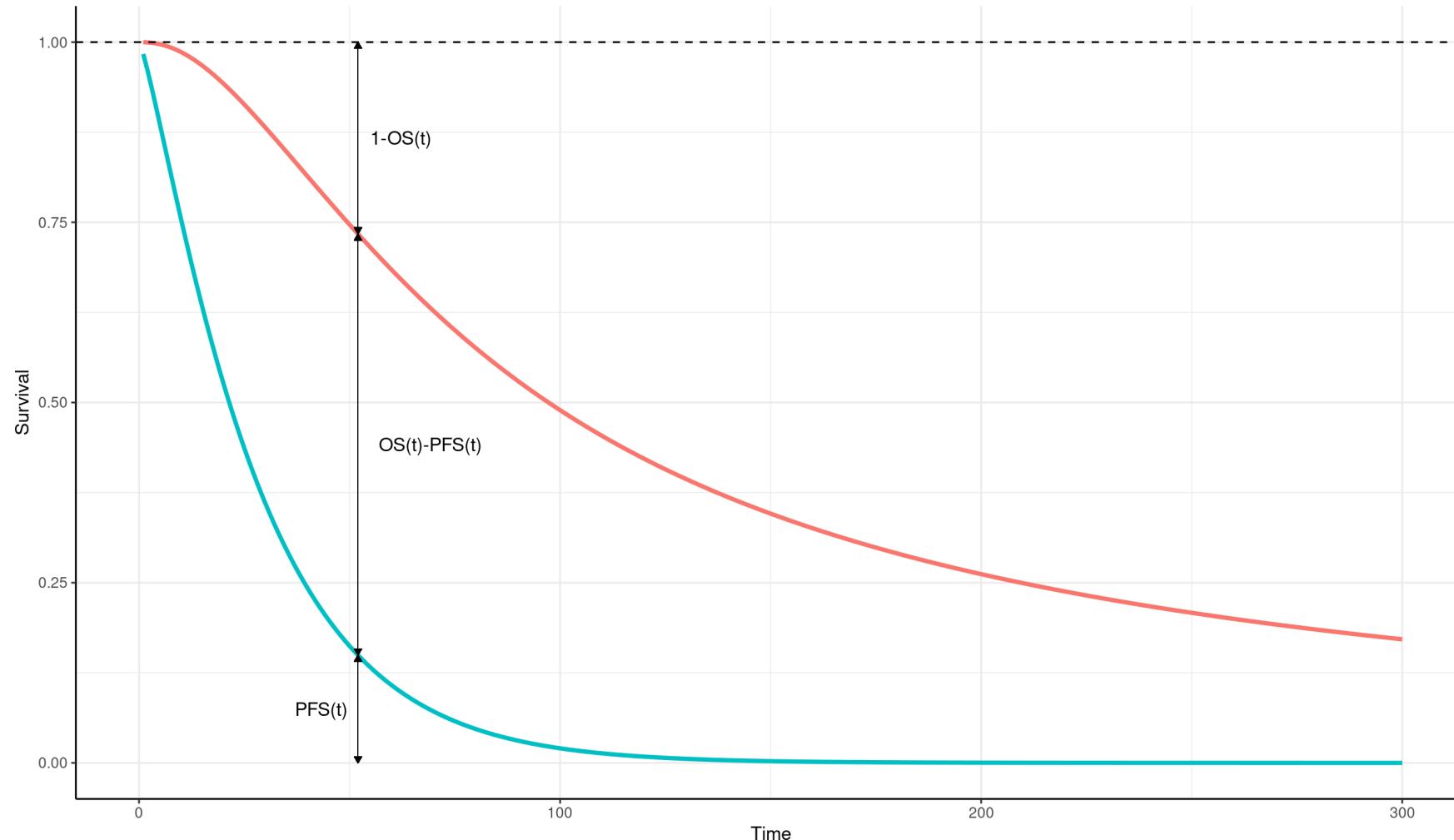
# Partitioned Survival Modelling (PSM)

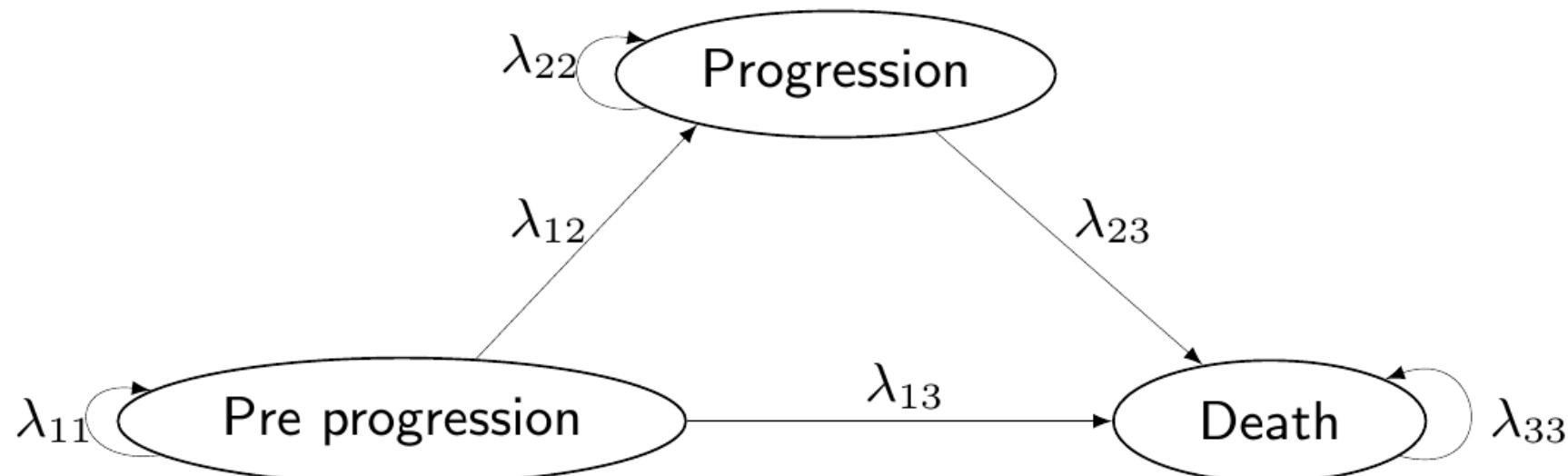
See  NICE DSU report



# Partitioned Survival Modelling (PSM)

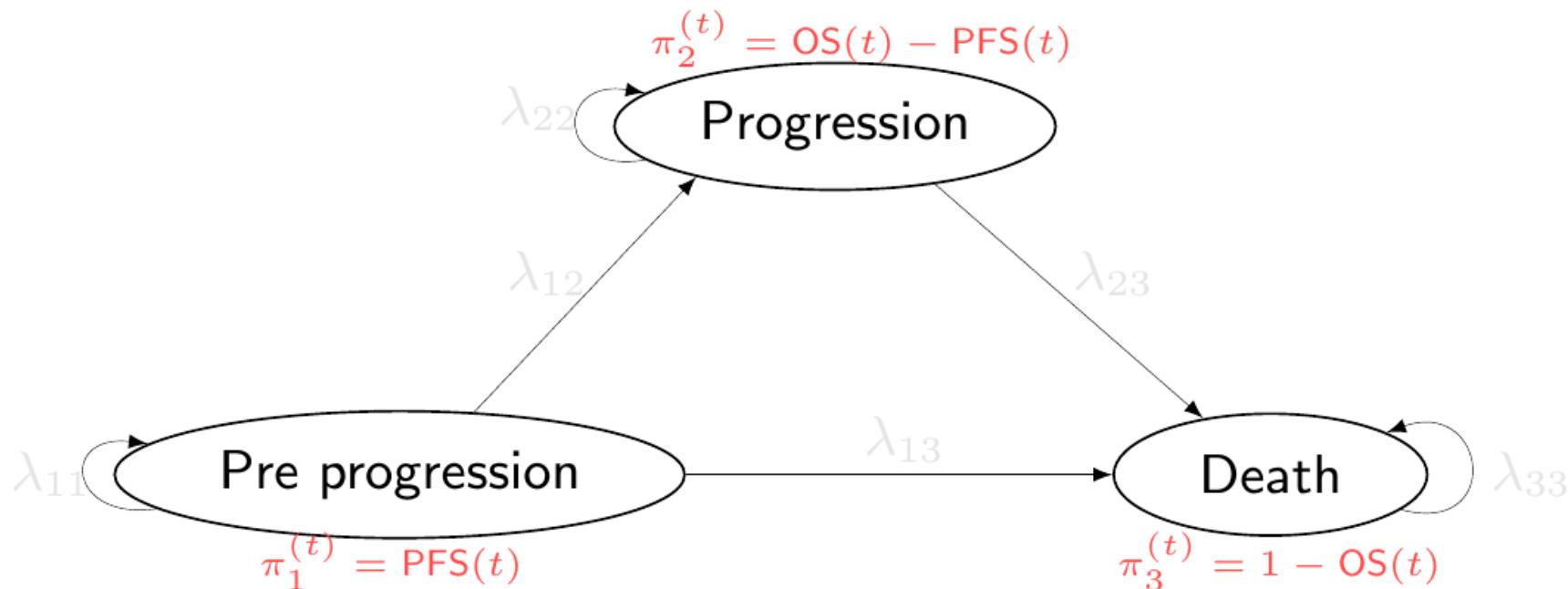
See  NICE DSU report





## Problem

- **Ideally**, want to estimate the transition probabilities  $\lambda$  to run the MM
- But, we're likely to only have access to (most likely digitised!) data on PFS/OS **separately**
  - PFS data record transition to either progression or death
  - Digitised OS data conflate  $\lambda_{12}$ ,  $\lambda_{13}$  and  $\lambda_{23}$



## (PSM) Solution

- Can estimate the **proportion of people in each state** at each time point
  - Basically run the simulation to attach costs & utilities to the number of individuals in each state at each time
  - **NB:** Partitioned survival analysis only applies for diseases where patients can only move **forward**
  - In a partitioned survival analysis, mortality is determined by time-to-death and not linked to concurring event

# Conclusions & further tools

- MMs are ubiquitous in health economic evaluation and HTA
- Once the transition probabilities have been estimated, the actual MM part is fairly easy
  - Based on matrix algebra – even Excel can handle it (in combination with statistical software for the main analysis...)
  - **But:** **much** more efficient to run the whole process through proper statistical software!

## Further tools

### heemod

- A general purpose package to run the whole MM process in R (can be linked to BCEA, but fundamentally a frequentist analysis)

### hesim

- A modular and computationally efficient R package for MMs; supports [doi cohort discrete time state transition models](#), [doi partitioned survival models](#) and [doi individual-level continuous time state transition models](#)

### Cost-effectiveness analysis in R using multi-state modelling

- A set of R scripts to run analyses based on individual level data (similar to the analysis shown in these slides + even more complex modelling structures)

### Bayesian Methods in Health Economics, chapter 5.4

 Library

 Book website (CRC)

 Book website

 Code

- (NB This includes a relatively old example, based on a [cohort discrete time state transition models](#), [doi](#), but it's useful to have a look to!)

 Next lecture