

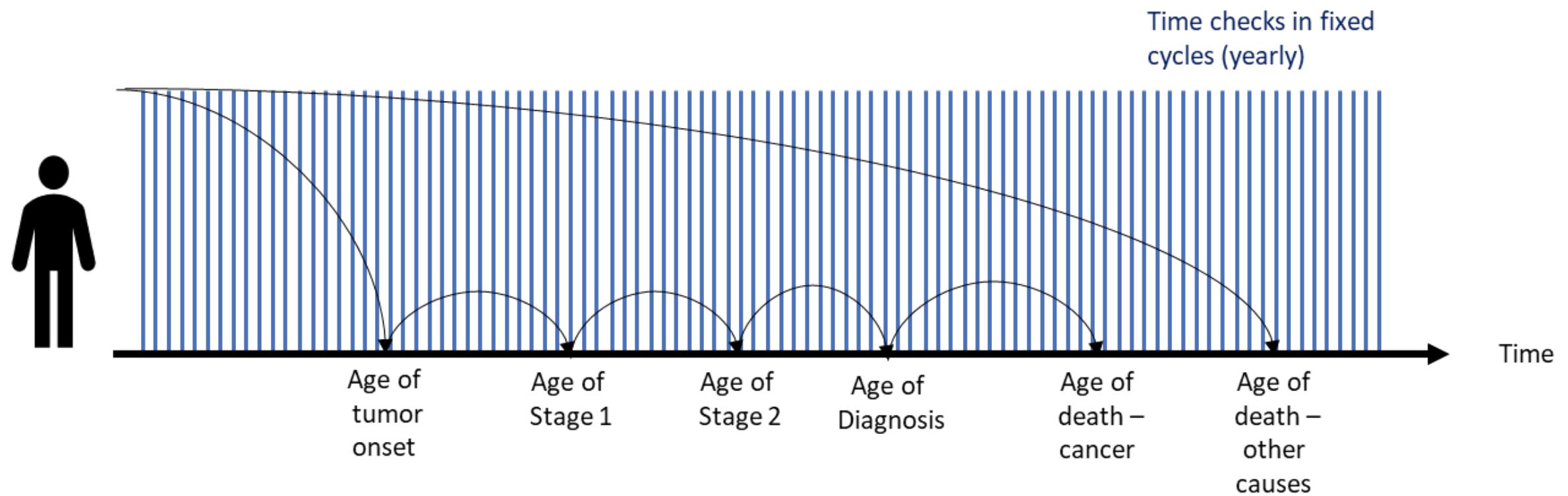
# Discrete-Event Simulation (DES) & Emulators

Day 5

# Discrete-Event Simulation (DES)

# Background

Individual-level **discrete-time simulation** models usually require sampling of which event happens each cycle.



# Background

Individual-level **discrete-event simulation (DES)** models usually require sampling times at which specific transitions or events could occur.

Animation courtesy of Carlos Pineda

# Background

Individual-level **discrete-time simulation** models usually require sampling of which event happens each cycle.

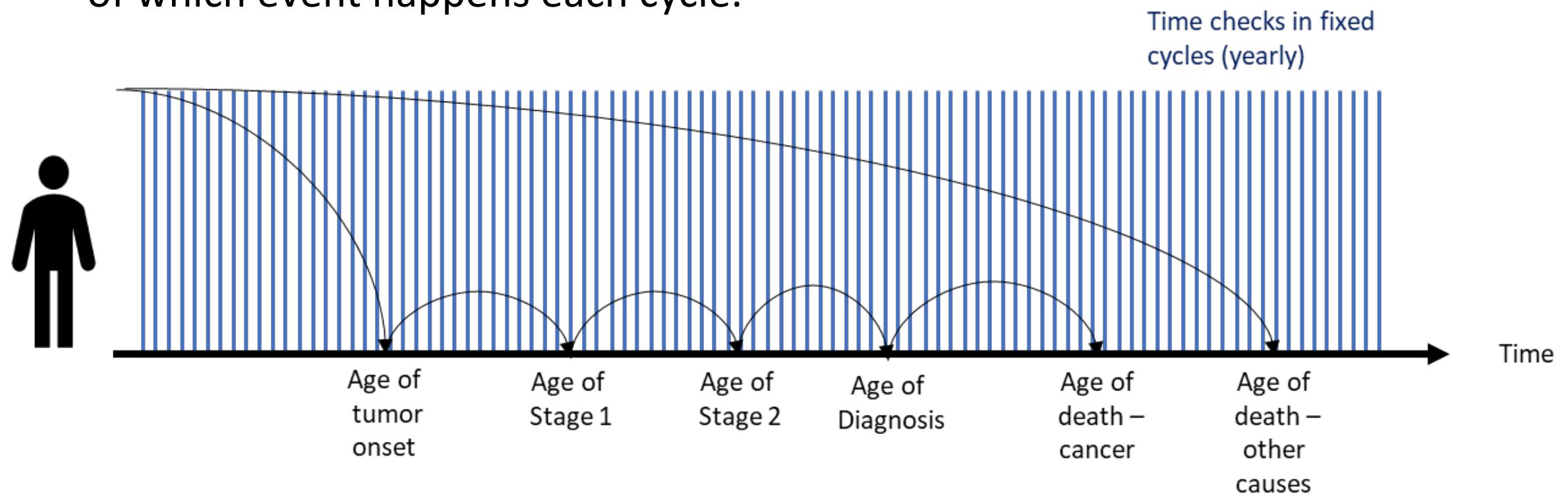


Figure courtesy of Carlos Pineda  
Stanford Health Policy

# Background

Individual-level **discrete-event simulation (DES)** models usually require sampling times at which specific transitions or events could occur.

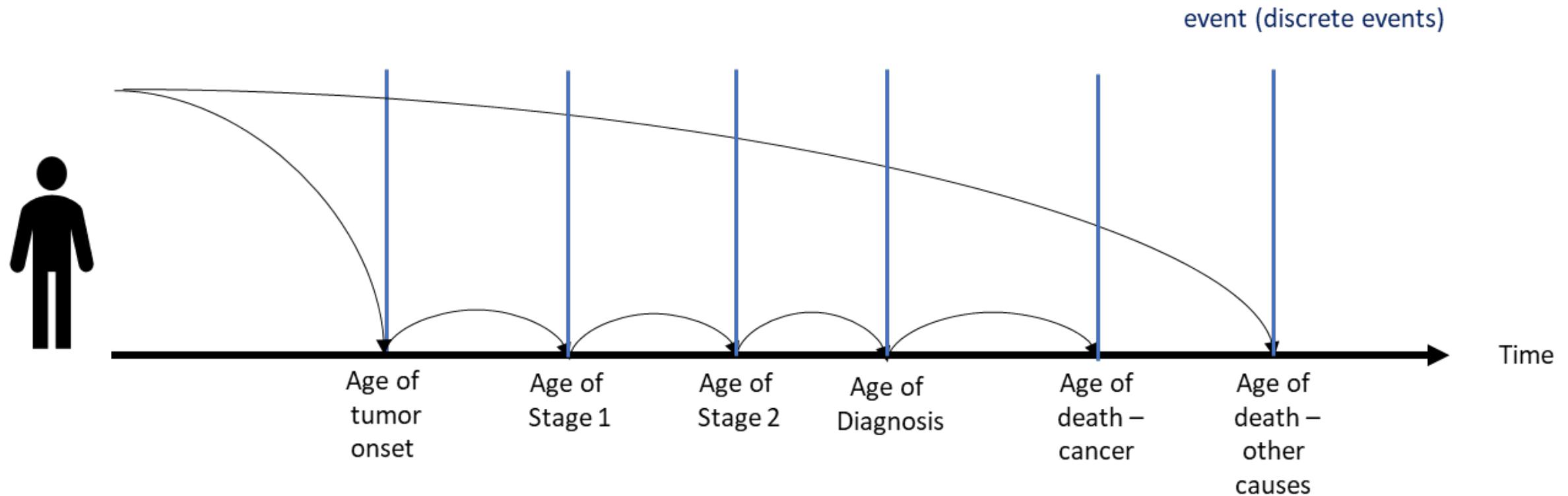
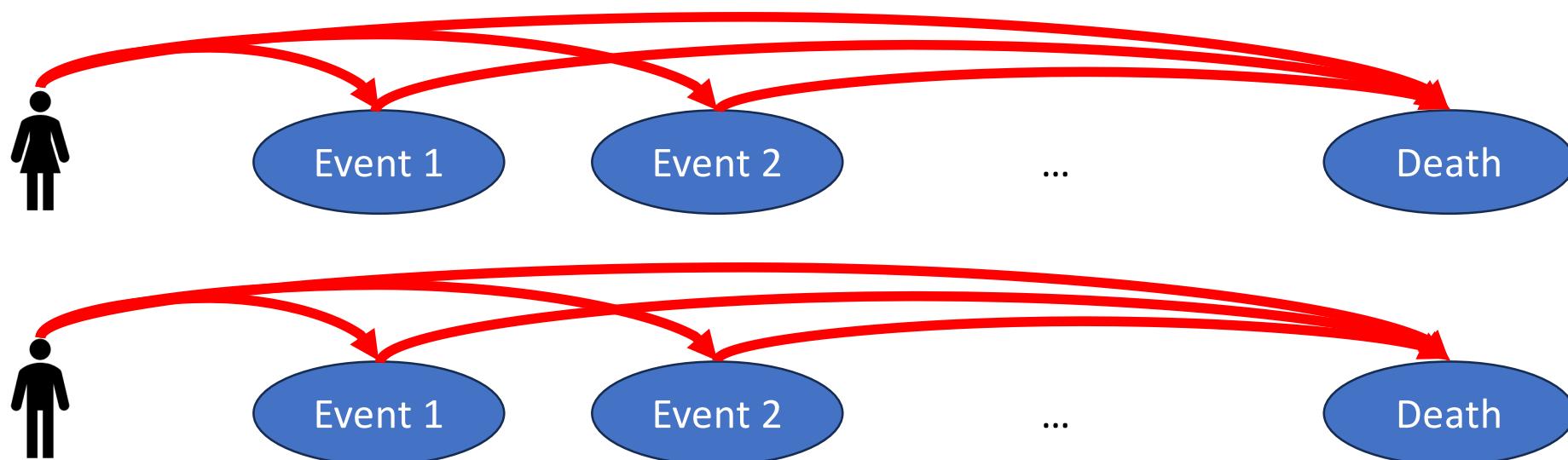


Figure courtesy of Carlos Pineda  
Stanford Health Policy

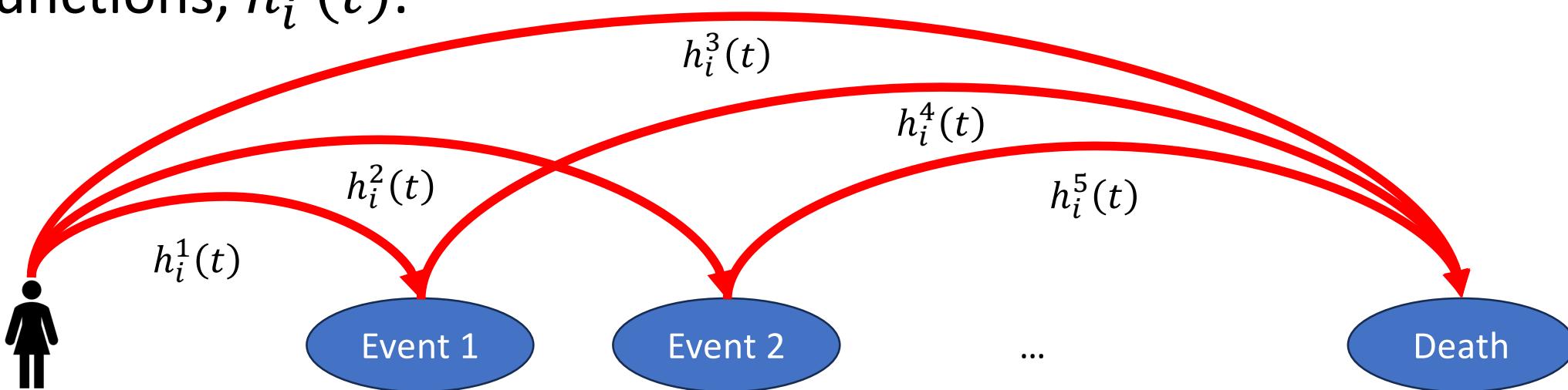
# Background

Individual-level discrete event simulation (DES) models, usually require sampling times at which specific transitions or events could occur.



# Background

Each of the  $k = 1, \dots, K$  possible transitions or events for each of the  $i = 1, \dots, N$  individuals are described by hazard functions,  $h_i^k(t)$ .



# Survival analysis in a nutshell

- Let  $T$  be the **random variable** of the time the event occurs
- **Survival** function: The probability of survival beyond time  $t$   
$$S(t) = \Pr(T > t) = 1 - F(t)$$
- **Mortality** function: Cumulative probability of having the event by time  $t$ , or the cumulative density function (CDF)

$$F(t) = \int_0^t f(s)ds ,$$

where  $f(t)$  is the probability density function (pdf).

# Survival analysis in a nutshell

- **Hazard** function: Instantaneous rate at which an event occurs at time  $t$  given survival until time  $t$

$$\lambda(t) = \frac{f(t)}{S(t)}$$

- **Cumulative hazard** function:

$$\Lambda(t) = \int_0^t \lambda(s)ds$$

# Survival analysis in a nutshell

- Survival as a function of cumulative hazard

$$S(t) = e^{-\Lambda(t)}$$

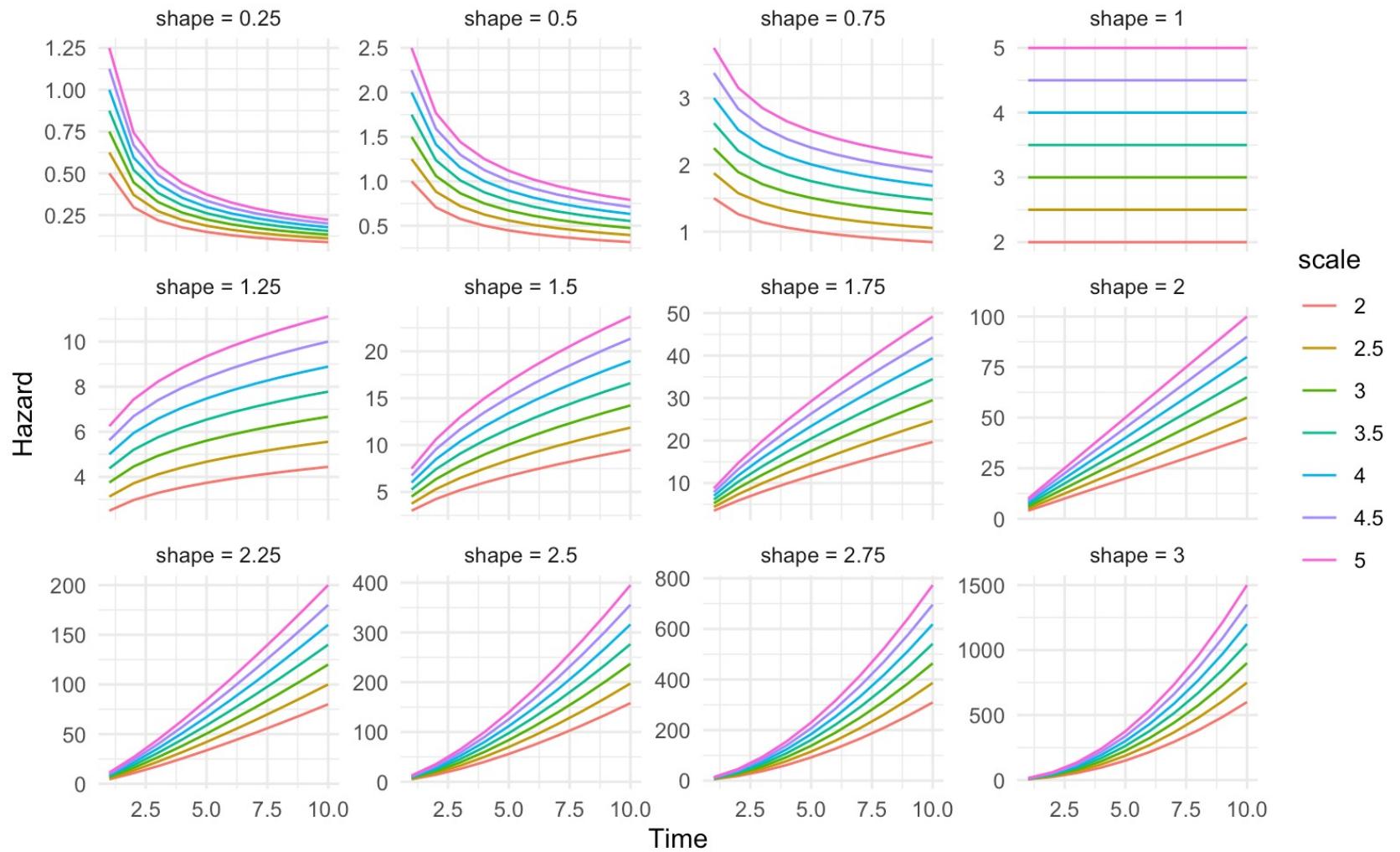
For a more detailed introduction to survival analysis:

Lee, E. T., & Wang, J. W. (2013). Functions of Survival Time. In Statistical Methods for Survival Data Analysis (4th ed., pp. 8–18). Wiley.

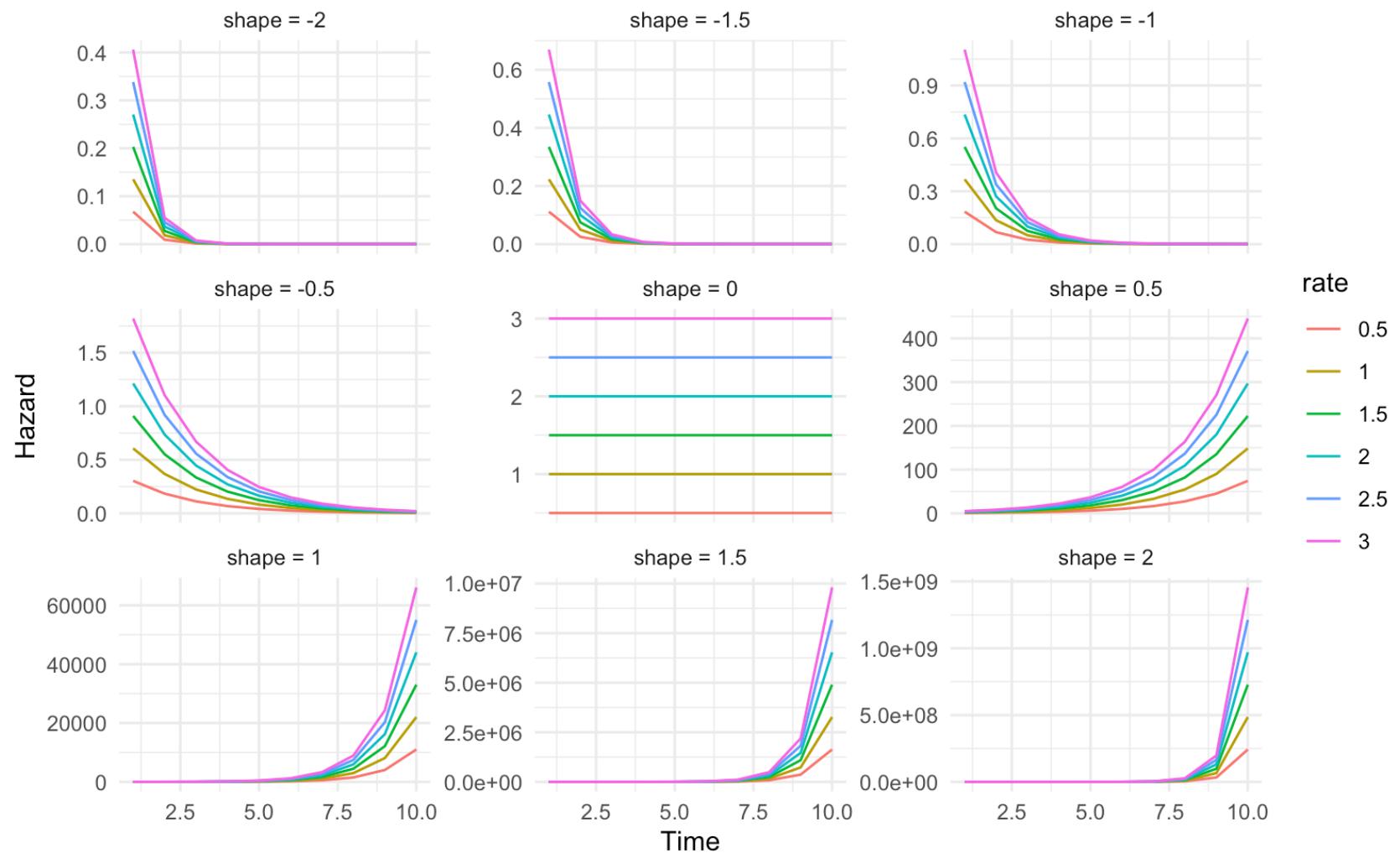
# Hazards

- A hazard function  $h(t)$  gives the *conditional rate* of an event
- It could be
  - Time-constant  $h(t) = h$
  - Time-dependent  $h(t)$ , or
- It could be a function of **covariates**
  - Time-independent covariates:  $h(t) = f(\mathbf{x}, t; \beta)$
  - Time-varying covariates (TVCs):  $h(t) = f(\mathbf{x}(t), t; \beta)$

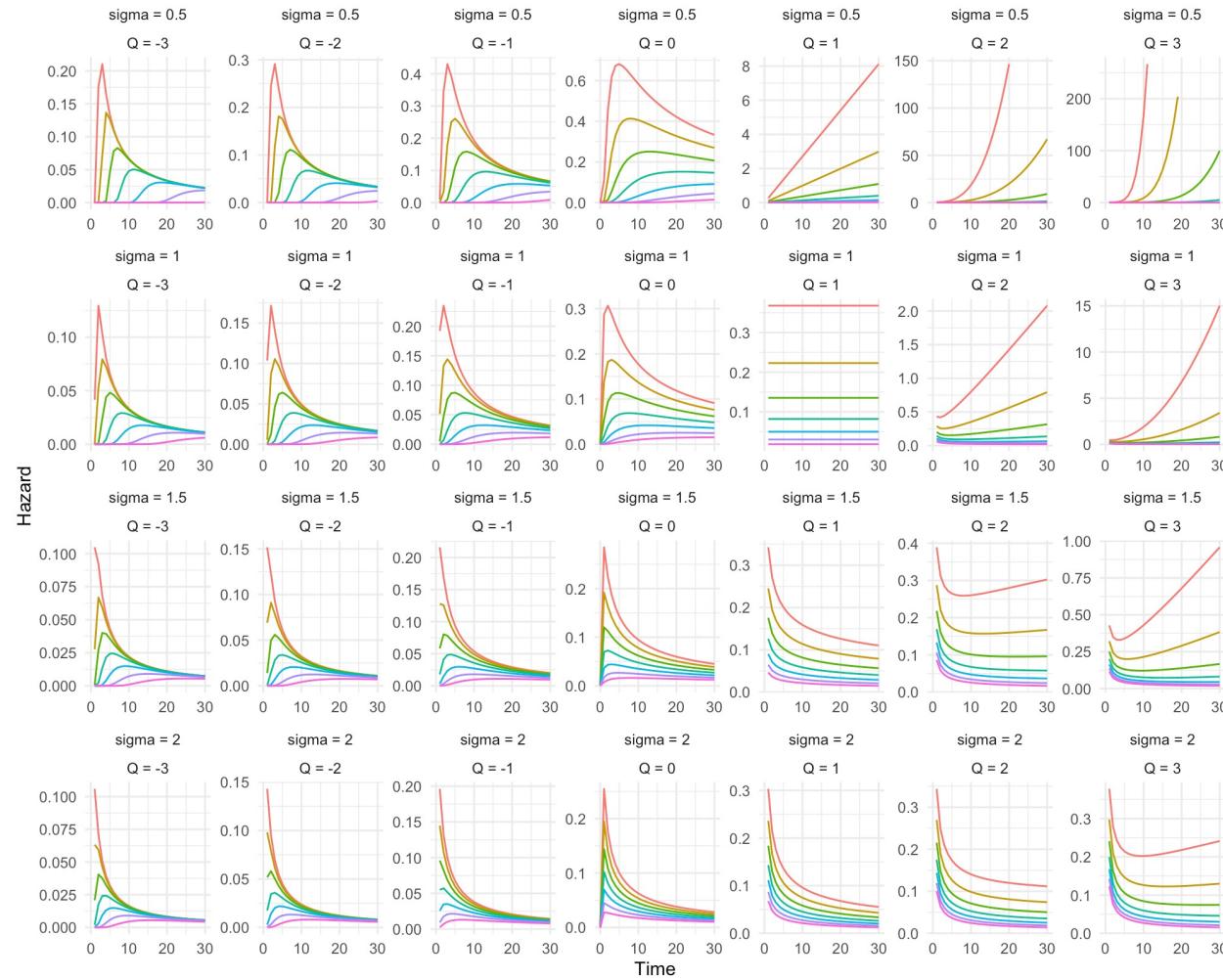
# Parametric hazards: Weibull



# Parametric hazards: Gompertz



# Parametric hazards: Generalized gamma



Source: [https://devinmcerti.com/2019/06/18/parametric\\_survival.html](https://devinmcerti.com/2019/06/18/parametric_survival.html)  
Stanford Freeman Spogli International

mu  
1  
1.5  
2  
2.5  
3  
3.5  
4

inford | Health Policy

# Sampling times to event

- From parametric distributions
  - Exponential
  - Gompertz
  - Weibull, etc.
- From nonparametric distributions
  - Kaplan-Meier
  - Nelson-Aalen
  - Life tables

# Sampling times to event

Sampling times to event  
following non-constant  
hazards is often done by  
drawing random values from  
**parametric distributions**

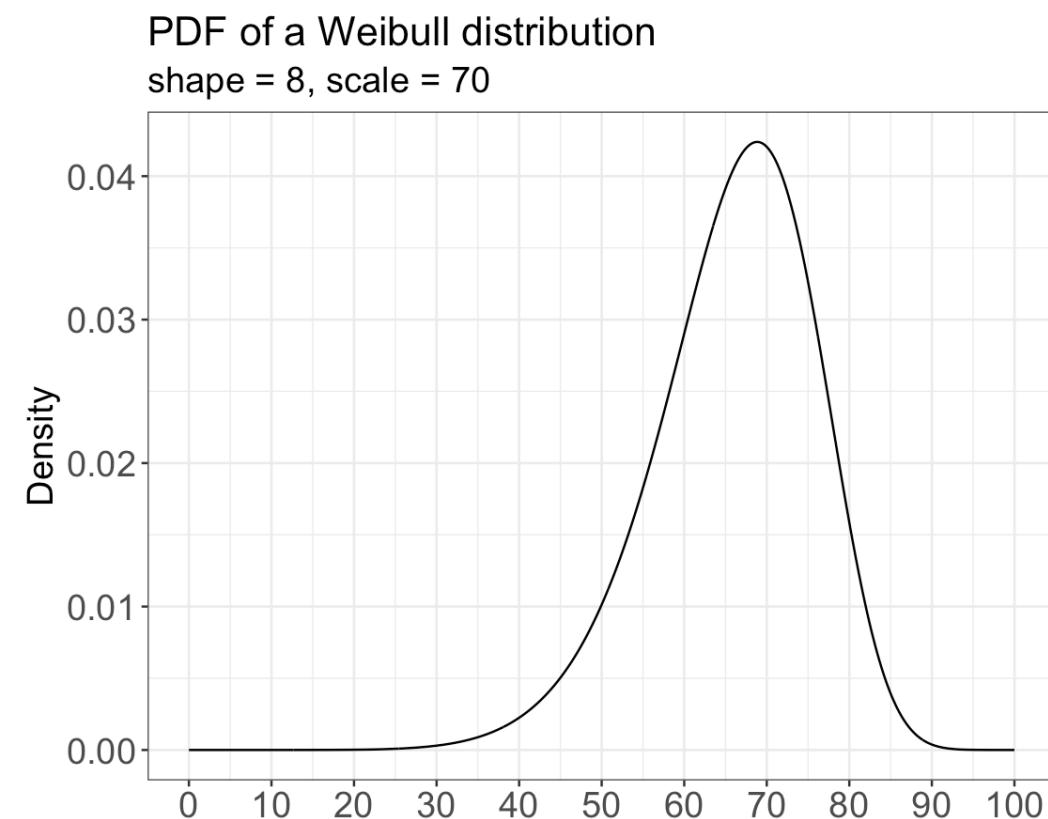
Weibull PDF

$$f(x; \lambda, k) = \frac{k}{\lambda} \left(\frac{x}{\lambda}\right)^{k-1}$$

Where  $\lambda$  is the scale and  $k$  is  
the shape

# Sampling times to event

Sampling times to event  
following non-constant  
hazards is often done by  
drawing random values from  
**parametric distributions**



# More in these works...

PLOS ONE

RESEARCH ARTICLE

## The nhpp package for simulating non-homogeneous Poisson point processes in R

Thomas A. Trikalinos<sup>1,2,3\*</sup>, Yuliia Sereda<sup>1</sup>



Original Research Article

## A Fast Nonparametric Sampling Method for Time to Event in Individual-Level Simulation Models

David U. Garibay-Treviño<sup>1</sup>, Hawre Jalal<sup>1</sup>, and Fernando Alarid-Escudero<sup>1</sup>



*Medical Decision Making*  
2025, Vol. 45(2) 205–213  
© The Author(s) 2025  
 Article reuse guidelines:  
[sagepub.com/journals-permissions](https://sagepub.com/journals-permissions)  
DOI: 10.1177/0272989X241308768  
 journals.sagepub.com/home/mdm



Stanford | Freeman Spogli Institute  
for International Studies

Stanford | Health Policy

# Sampling events

One **individual** at a time

-> **inefficient** in high-level languages like R or Python



# Sampling events

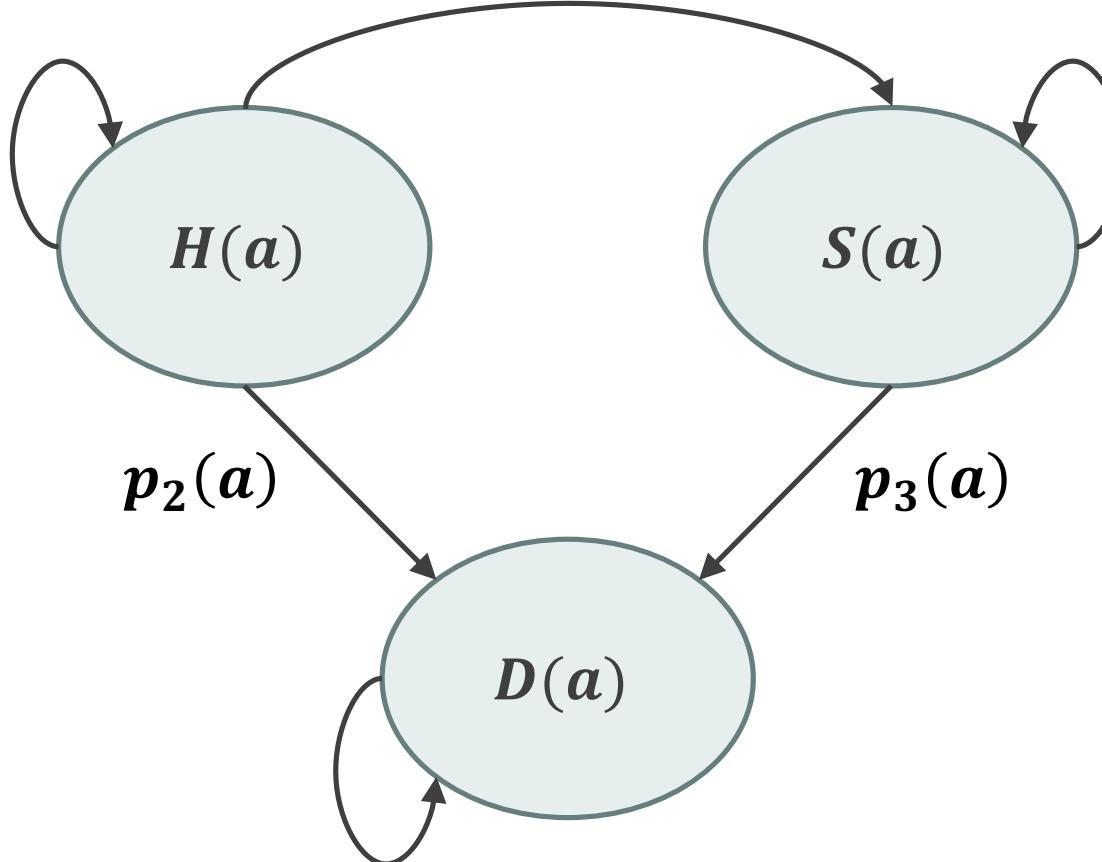
One **event** at a time

-> **efficient** in high-level languages like R or Python



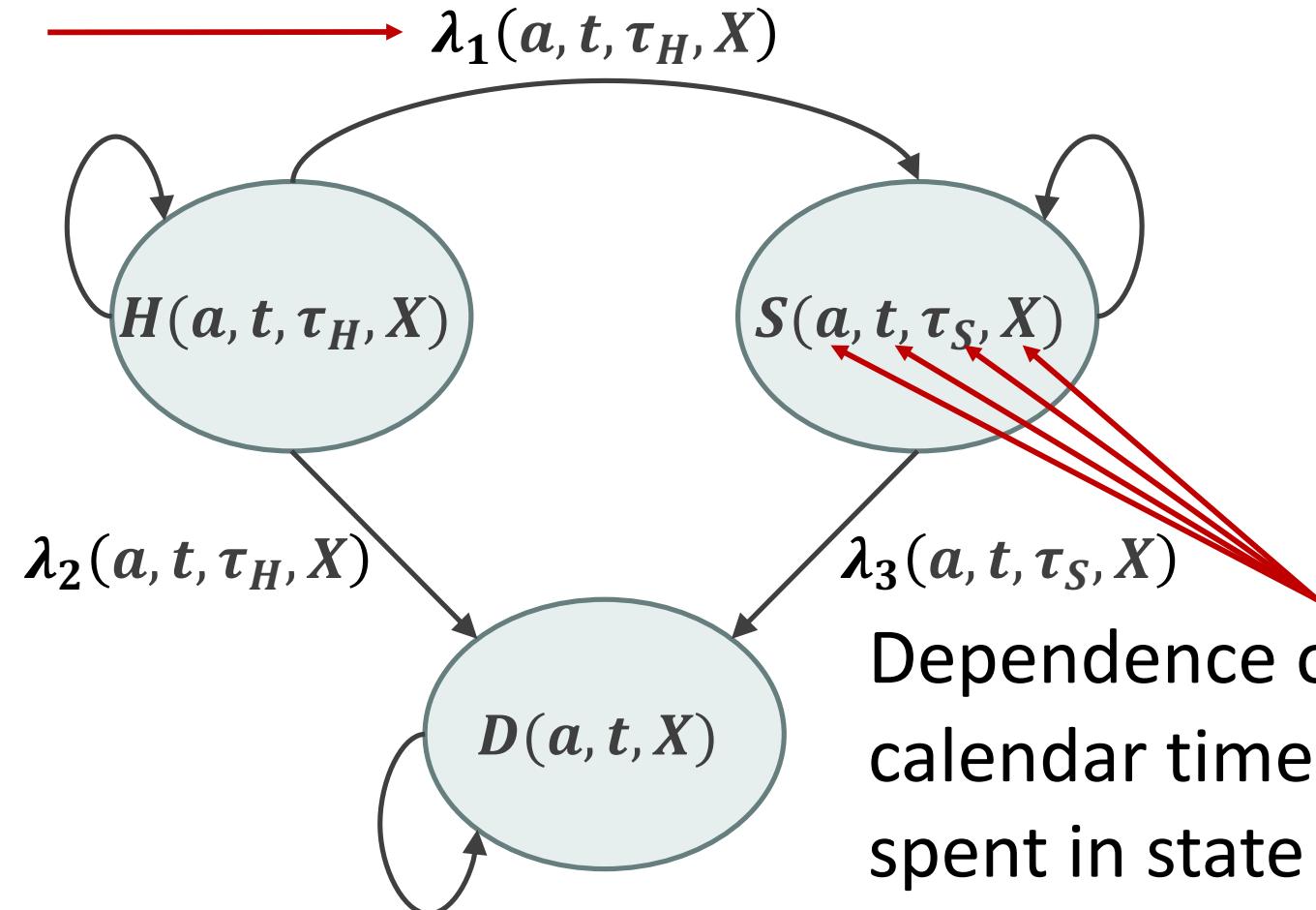
# State Transition Diagram (STD) of a Cohort Model

Probabilities  $\longrightarrow p_1(a)$



# State Transition Diagram (STD) of a DES Model

Hazards/rates

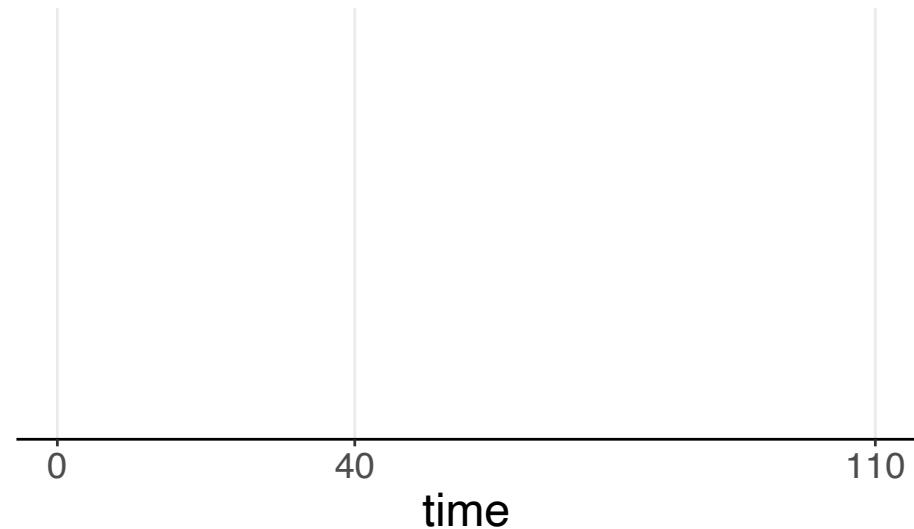


Dependence on age ( $a$ ),  
calendar time ( $t$ ), time  
spent in state  $k$  ( $\tau_k$ ), and  
covariates ( $X$ )

# Graphical notation

The time horizon of the simulation

- Stop the simulation at 110



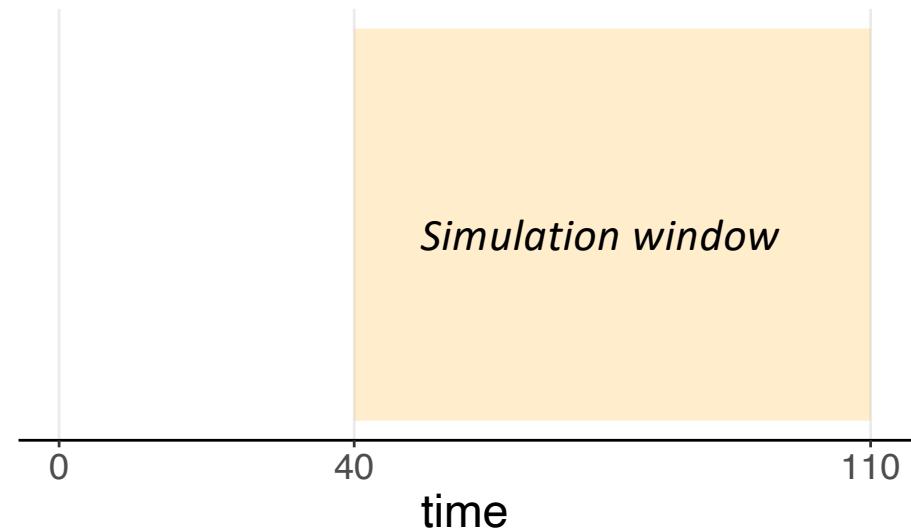
*All times in years*

# Graphical notation

We are interested in the interval from 40 to 110

- Spawn cancer-free at 40
- Stop the simulation at 110

All our cancer-related events may occur in the shaded window.



# Graphical notation: Type of events

1. Exactly one event
2. At most one event
3. Zero, one, or more events

*Simulation window*

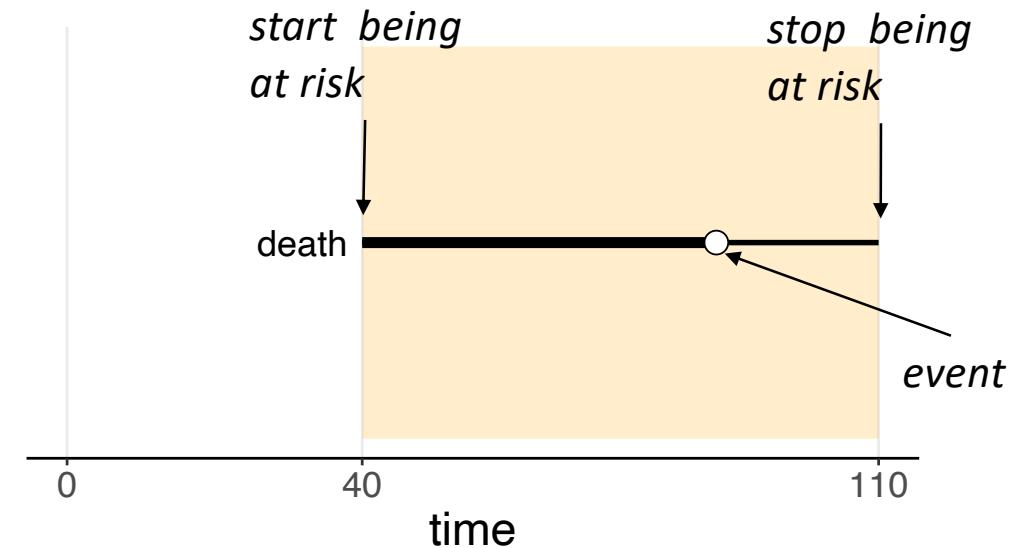
# Graphical notation: Exactly one event

Some events shall happen **exactly once** in the interval of interest.

We use black color for such processes.

Example:

- death from all causes



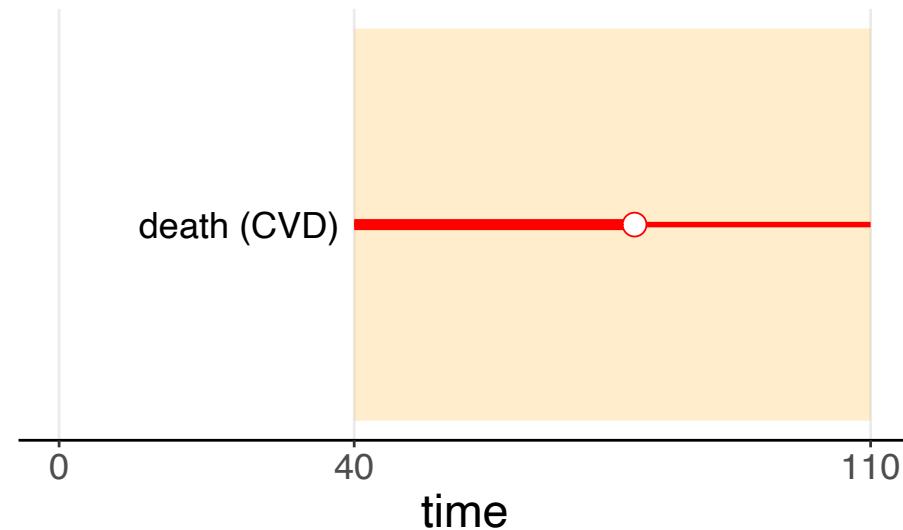
# Graphical notation: At most one event

Some events shall happen **at most once** in the interval of interest.

Note, color red.

Example:

- Death from cardiovascular disease (CVD) occurred at age 78



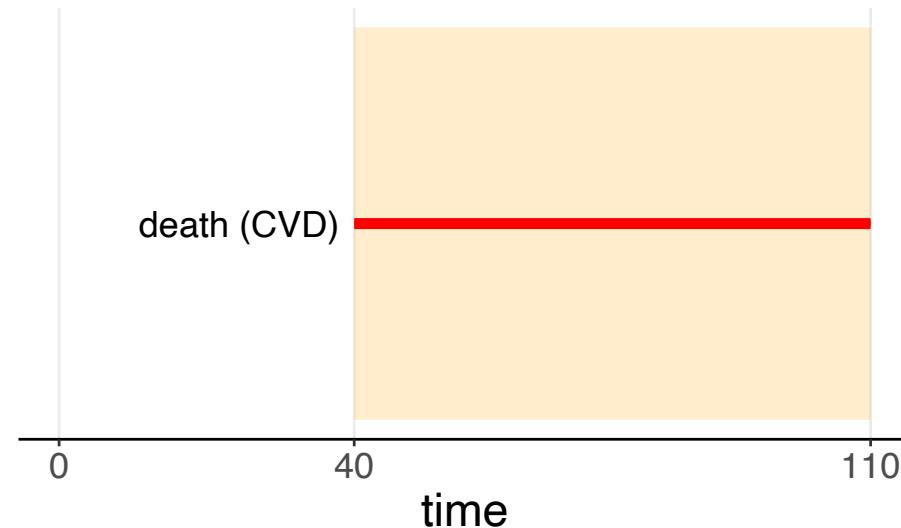
# Graphical notation: At most one event

Some events shall happen **at most once** in the interval of interest.

Note, color red.

Example:

- No death throughout the at-risk interval



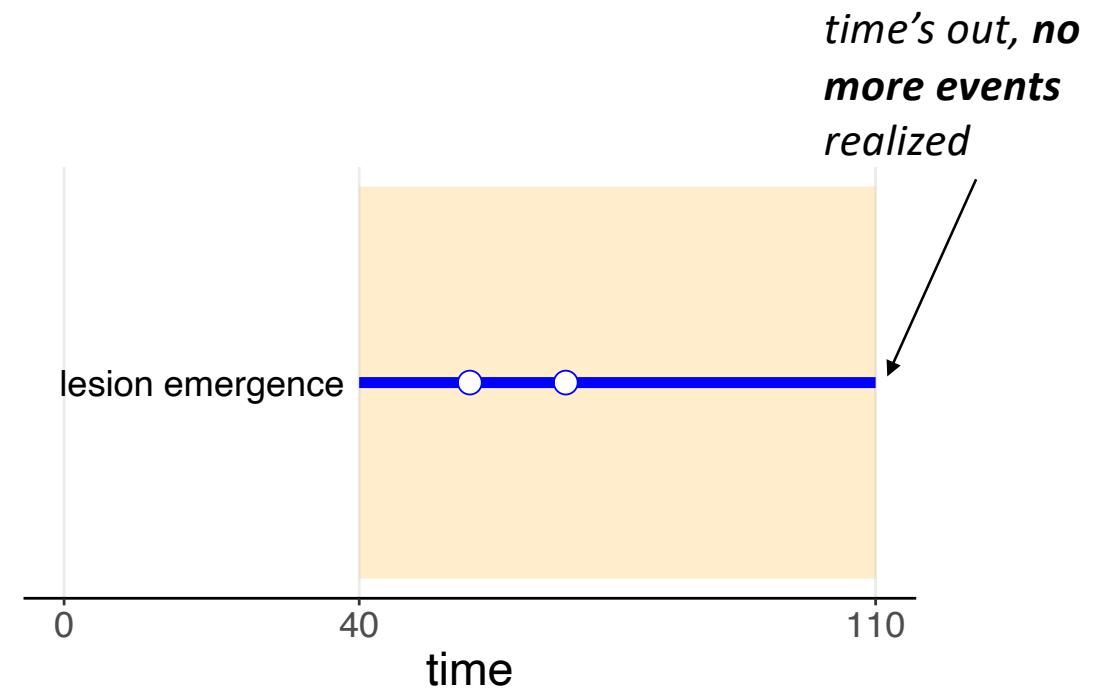
# Graphical notation: Zero, one, or more events

Some events may happen **zero, one or more times** in the interval of interest.

Note, color blue.

Example:

- Occurrence of lesions at 55 and 68 years



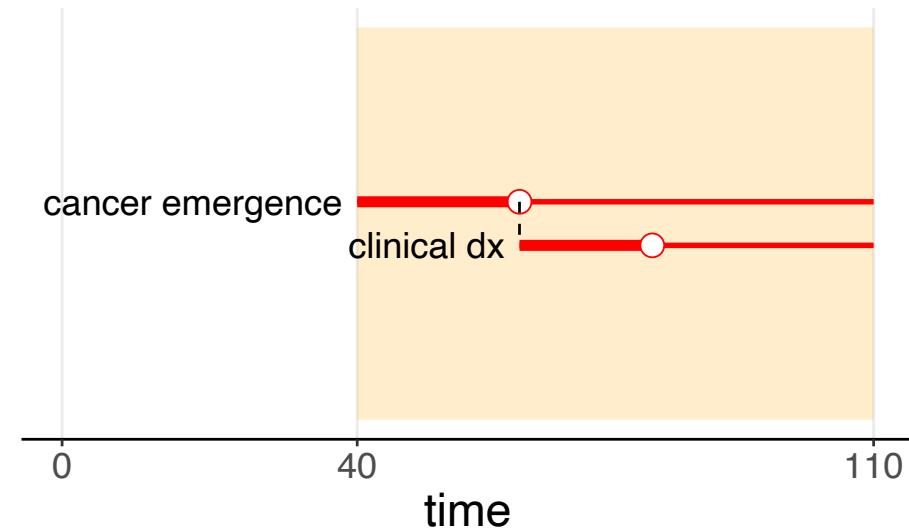
# Graphical notation: Chained events (in series)

For chained processes, the next one starts once the preceding one realizes an event.

Example:

- Clinical cancer diagnosis happens at 80, but the process starts only after cancer has emerged at 62

*imagine simulating  
the first row first, etc.*

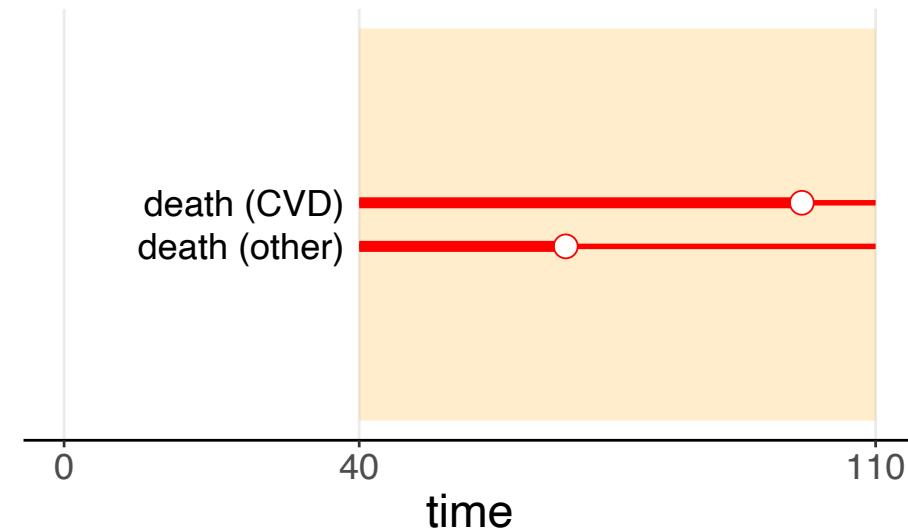


# Graphical notation: Competing events (parallel)

Competing event processes run parallel to each other.

Example:

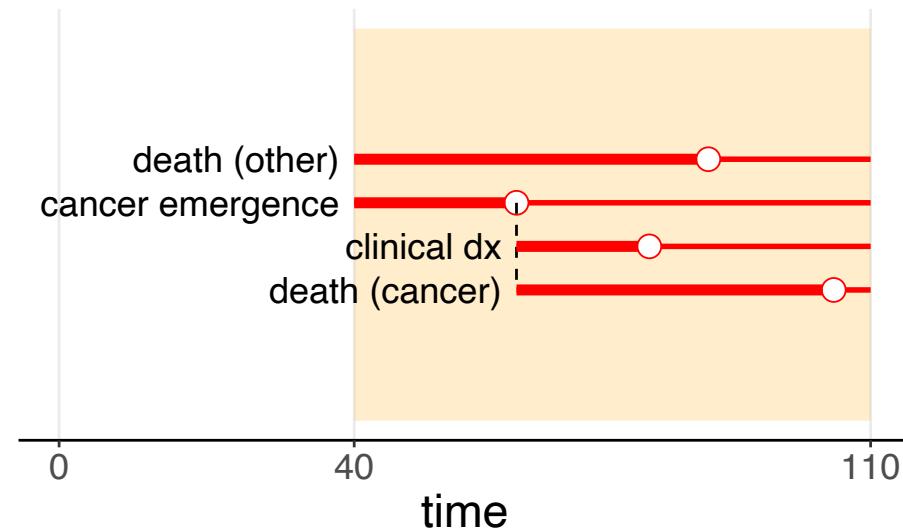
- Death from CVD at 100, death from non-CVD causes at 68
- The age of all cause death is the earliest occurring event, if any (no guaranteed death in interval)



# A simple DES model

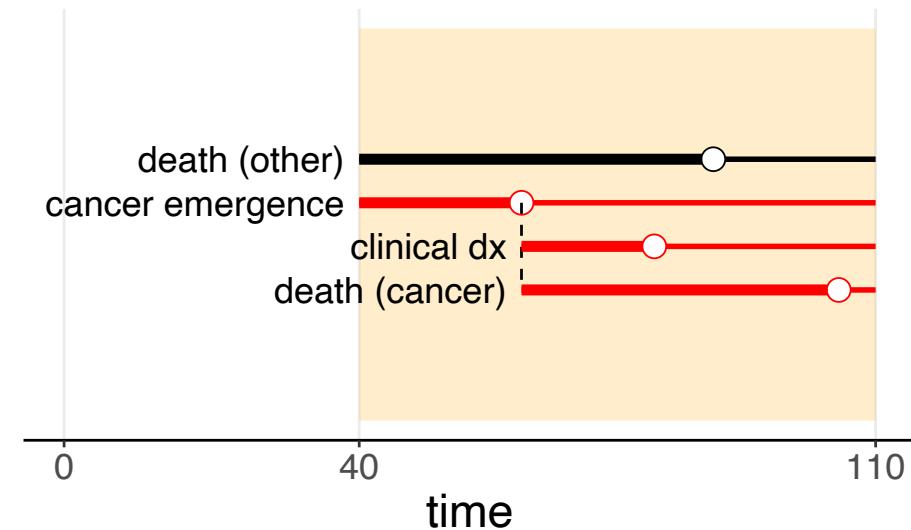
A DES model comprises the black, red, and blue processes, connected in series or in parallel, with proper accounting of start and stop ages.

- What does the modeler assume in this example?

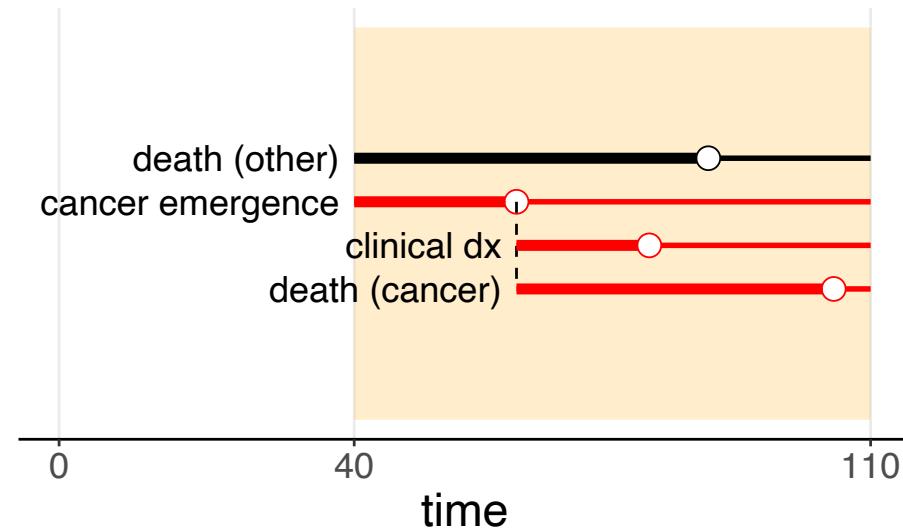
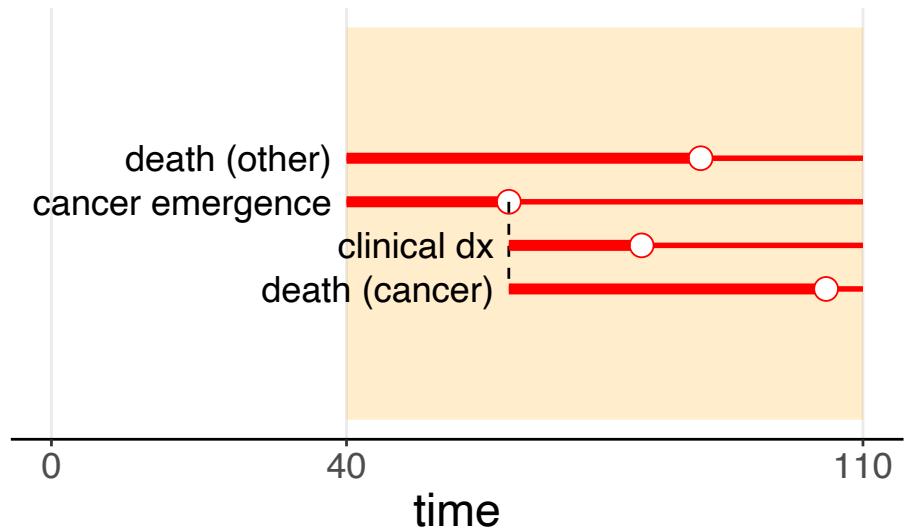


# Another simple DES model

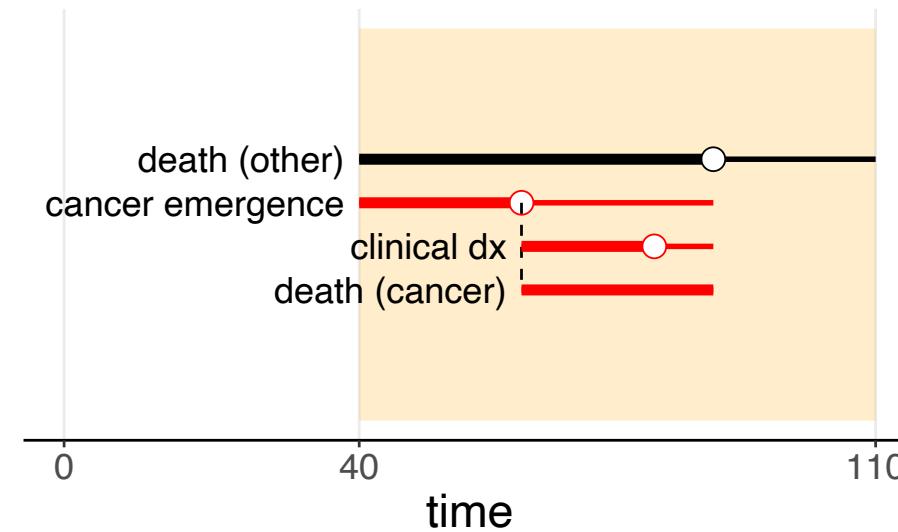
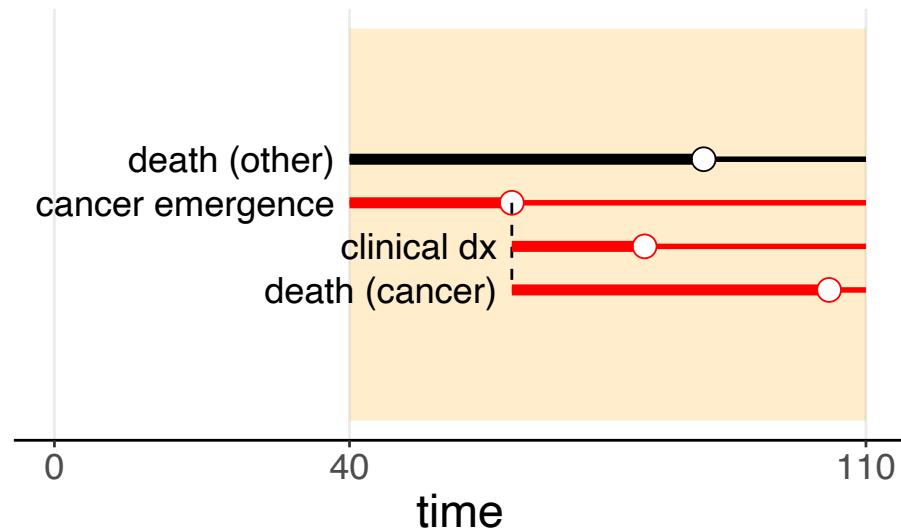
- What does the modeler assume in this example?



# The two examples side by side



# Cancer death: what at-risk interval was chosen?



# The building blocks of a DES



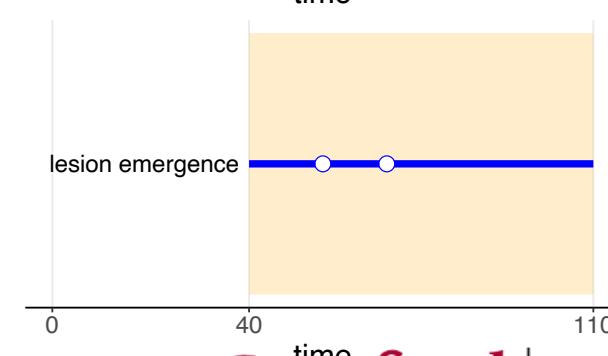
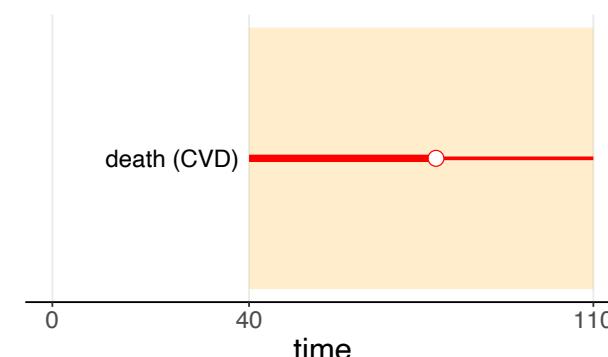
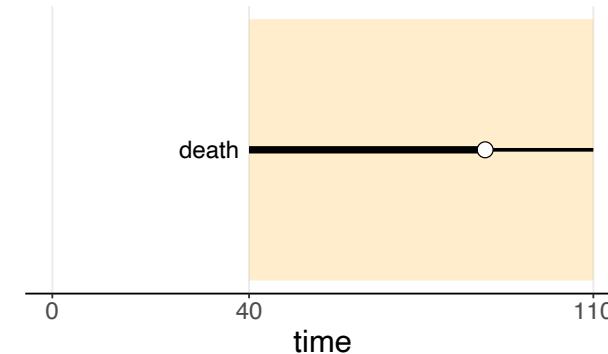
Events that happen exactly once



Events that happen 0 or 1 times



Events that happen 0, 1, ... times

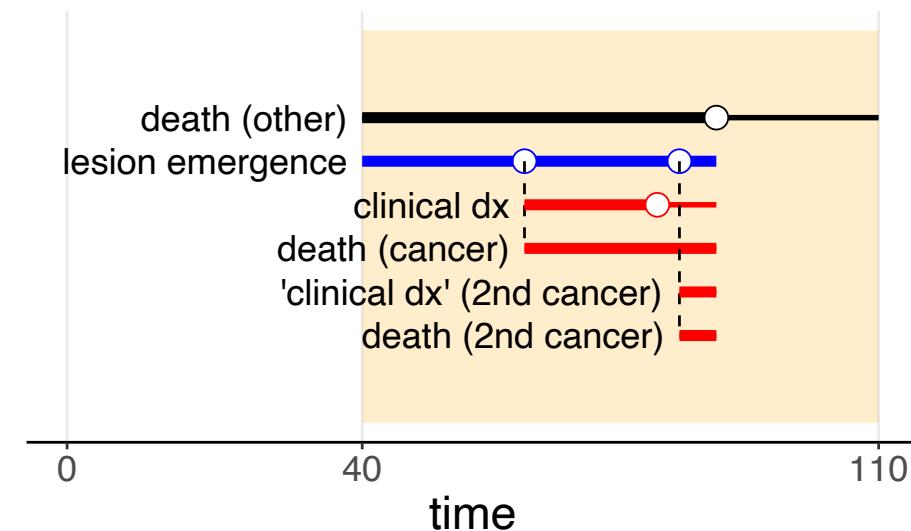


# What would a model with multiple tumors look like?

# A model with multiple tumors

Many architectures are possible.

What are the risk intervals for each event process?



# A tutorial on these works

# medRxiv

THE PREPRINT SERVER FOR HEALTH SCIENCES

HOME

Soon to be published in PharmacoEconomics!

 [Follow this preprint](#)

## **Discrete-Event Simulation Modeling Framework for Cancer Interventions and Population Health in R (DESCIPHR): An Open-Source Pipeline**

 Selina Pi,  Carolyn M. Rutter,  Carlos Pineda-Antunez,  Jonathan H. Chen,  
 Jeremy D. Goldhaber-Fiebert,  Fernando Alarid-Escudero

**doi:** <https://doi.org/10.1101/2025.05.12.25327470>

<https://www.medrxiv.org/content/10.1101/2025.05.12.25327470v2>

**Stanford** | Freeman Spogli Institute  
for International Studies

**Stanford** | Health Policy

# Notation

- $i = 1, \dots, N$ : individual
- $N$ : the number of individuals
- $k = 1, \dots, K$ : non-dead state
- $K$ : the number of non-dead states
- $\tau_k^i$ : years spent by individual  $i$  in state  $k$
- $L_i$ : Life years lived by individual  $i$
- $E[L]$ : Expected life years
- $Q_i$ : Quality-adjusted life years (QALYs) for individual  $i$
- $E[Q]$ : Expected QALYs
- $u_k$ : Utility of spending a year in state  $k$ .
- $C_i$ : Cost by individual  $i$
- $E[C]$ : Expected costs
- $c_k$ : Cost of spending a year in state  $k$
- $O_i$ : Outcome for individual  $i$
- $E[O]$ : Expected outcome
- $\rho_O$ : Exponential discount rate for outcome  $O$

# Calculating outcomes

- Life years ( $L_i$ )

$$L_i = \sum_{k=1}^K 1 \cdot \tau_k^i,$$

where  $K$  is the number of non-dead states, and  $\tau_k^i$  the years spent by individual  $i$  in state  $k$ ,

- Expected life years ( $E[L]$ )

$$E[L] = \frac{1}{N} \sum_{i=1}^N L_i,$$

where  $N$  is the number of individuals.

# Calculating outcomes

- Quality-adjusted life years ( $Q_i$ )

$$Q_i = \sum_{k=1}^K u_k \cdot \tau_k^i,$$

where  $K$  is the number of non-dead states,  $\tau_k^i$  the time spent by individual  $i$  in state  $k$ , and  $u_k$  the utility of spending a year in state  $k$ .

- Expected quality-adjusted life years ( $E[Q]$ )

$$E[Q] = \frac{1}{N} \sum_{i=1}^N Q_i,$$

where  $N$  is the number of individuals.

# Calculating outcomes

- Costs ( $C_i$ )

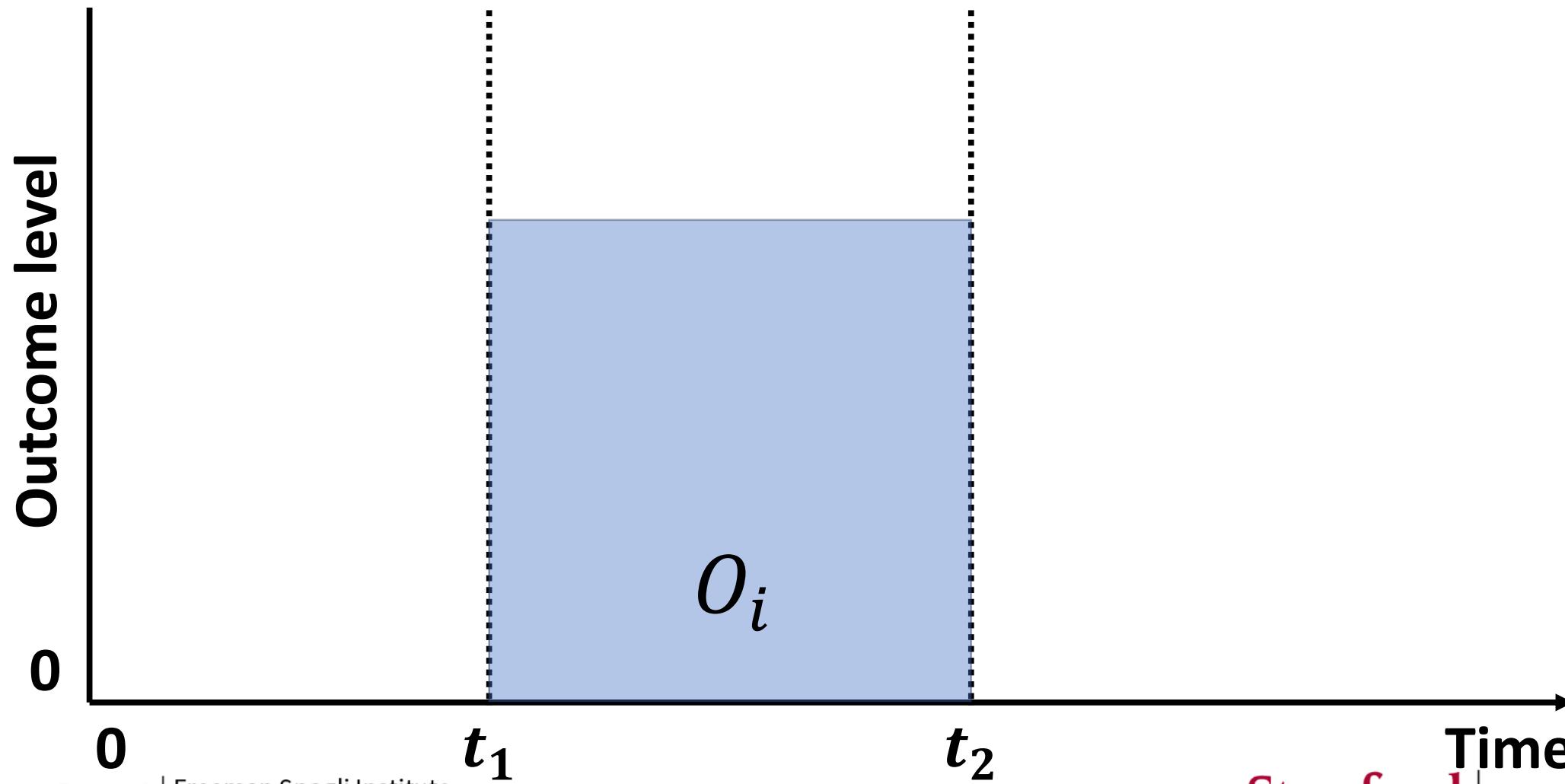
$$C_i = \sum_{k=1}^K c_k \cdot \tau_k^i,$$

where  $K$  is the number of non-dead states,  $\tau_k^i$  the time spent by individual  $i$  in state  $k$ , and  $c_k$  the cost of spending a year in state  $k$ .

- Expected costs ( $E[C]$ )

$$E[C] = \frac{1}{N} \sum_{i=1}^N C_i$$

# Cumulative outcomes



# Discounted outcomes

- Let  $O_i$  be a cumulative outcome for individual  $i$  over a period  $[t_1, t_2]$ , and  $\rho_O$  the exponential discount rate for outcome  $O$ , then the exponentially discounted cumulative outcome  $\tilde{O}_i$  is

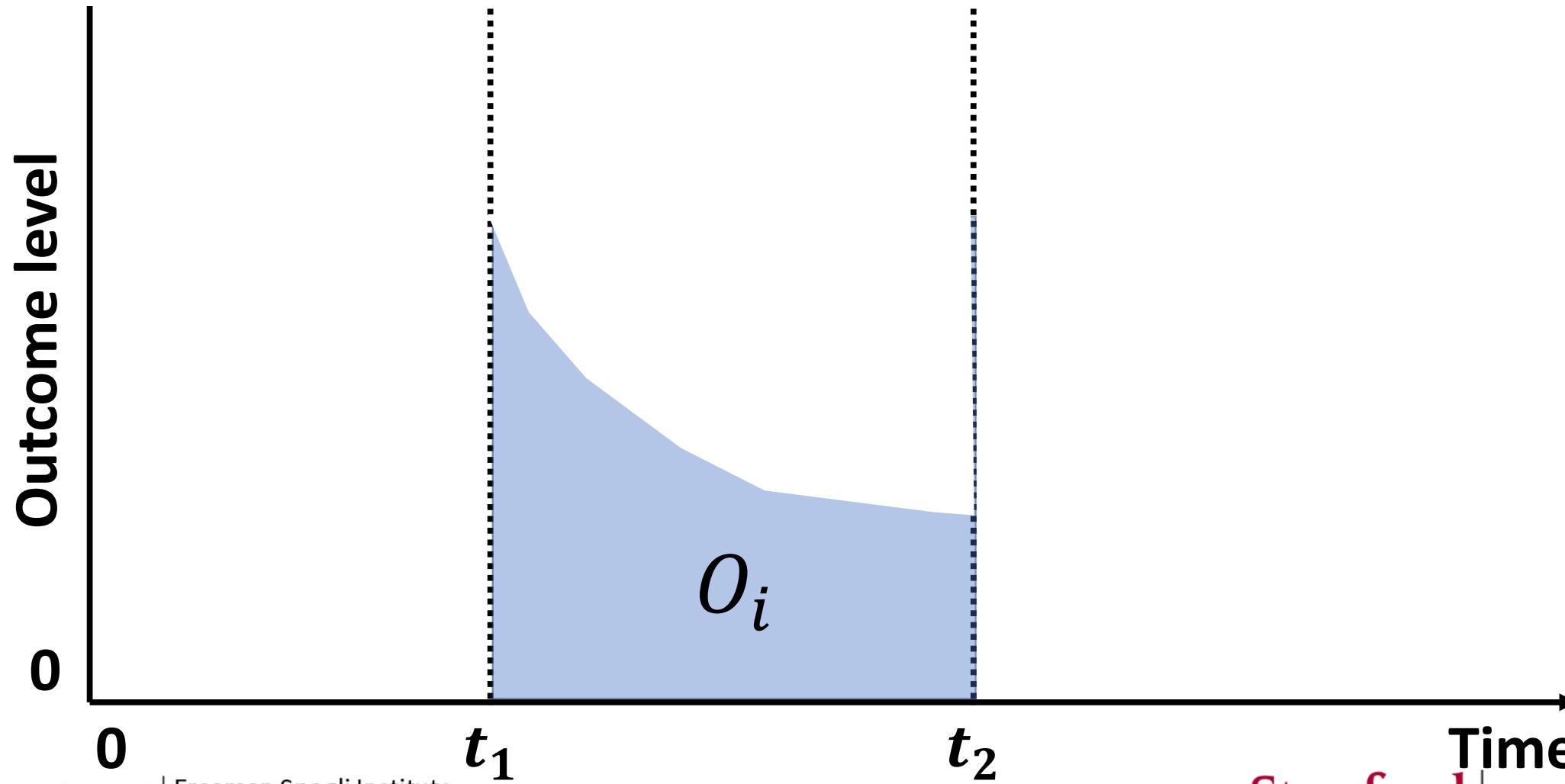
$$\tilde{O}_i = \int_{t_1}^{t_2} O_i e^{-\rho_O t} dt = O_i \frac{(e^{-\rho_O t_1} - e^{-\rho_O t_2})}{\rho_O}$$

- The expected discounted cumulative outcome ( $E[\tilde{O}]$ )

$$E[\tilde{O}] = \frac{1}{N} \sum_i^N \tilde{O}_i,$$

where  $N$  is the number of individuals.

# Cumulative outcomes



# A tutorial on these works



THE PREPRINT SERVER FOR HEALTH SCIENCES

HOME

Follow this preprint

## **A Tutorial on Discrete Event Simulation Models in R using a Cost-Effectiveness Analysis Example**

Mauricio Lopez-Mendez, Jeremy D. Goldhaber-Fiebert, Fernando Alarid-Escudero

**doi:** <https://doi.org/10.1101/2025.05.15.25327635>

<https://www.medrxiv.org/content/10.1101/2025.05.15.25327635v2>

# DES R Lab

**Stanford** | Freeman Spogli Institute  
for International Studies

**Stanford** | Health Policy

# Emulator-Based Sensitivity Analysis

# Motivation

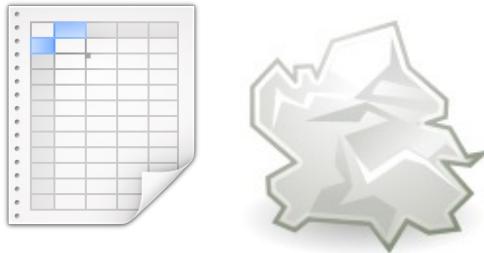
- Conducting sensitivity analyses could be time-consuming
  - Re-running the model
- Calibration (particularly, Bayesian) is computationally demanding
- Reproducing prior analyses is often challenging

# Computing Time Example

- Base-case analysis = 1 millisecond
- DSA (1 parameter) = a few milliseconds
- PSA 10000 = 10sec
- PSA 10,000 x MCS 10,000 = 100,000,000 sec = 28hrs
- PSA can take a lot of time!
  - Rich data
  - Not fully utilized

# Usual Fate of PSA

PSA



# Usual Fate of PSA

PSA



# Motivation

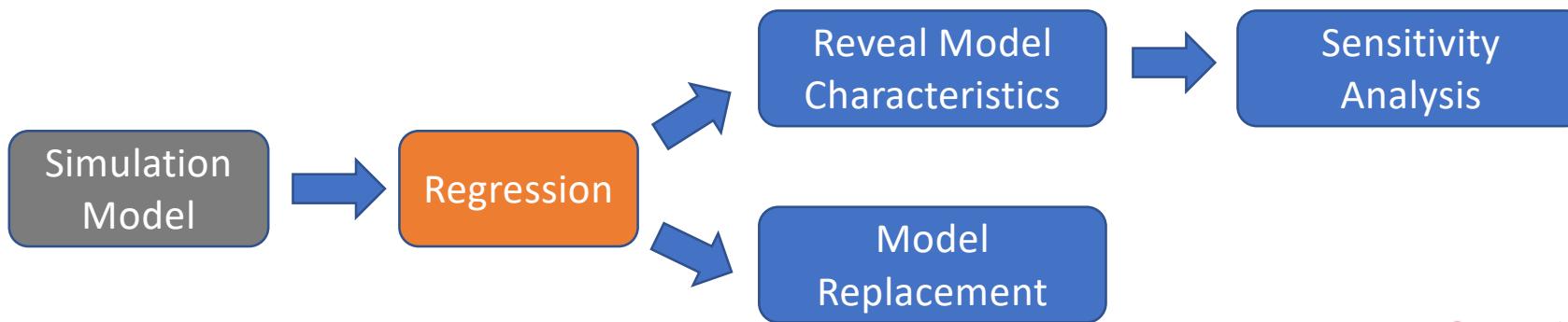
- Bring back the PSA
- Use metamodeling to generate a panel of SA results from the PSA dataset
- Offer an open-source implementation that can be applied and extended to a wide range of simulation models.

# What is Regression Metamodeling?

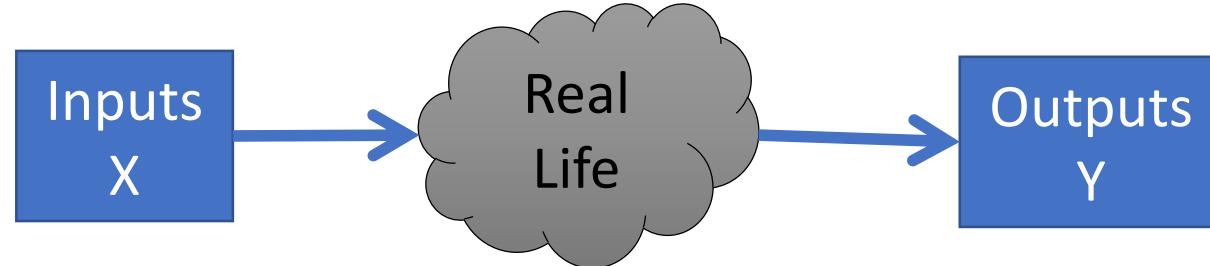
- Familiar with regression analysis



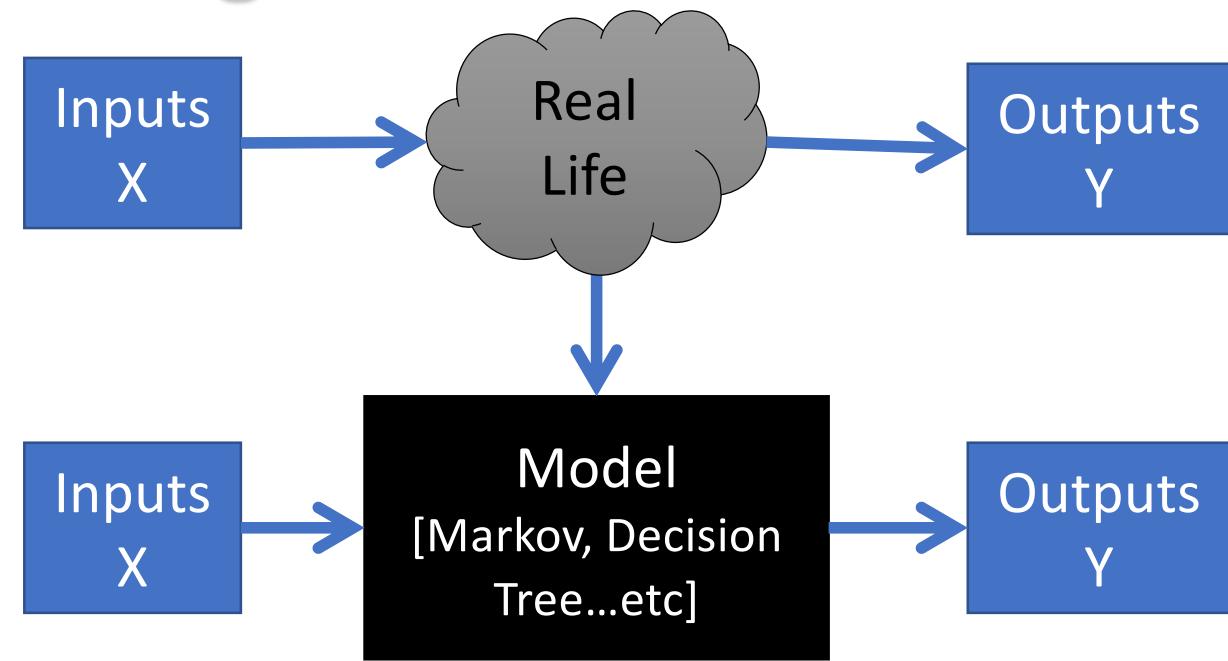
- Regression metamodeling



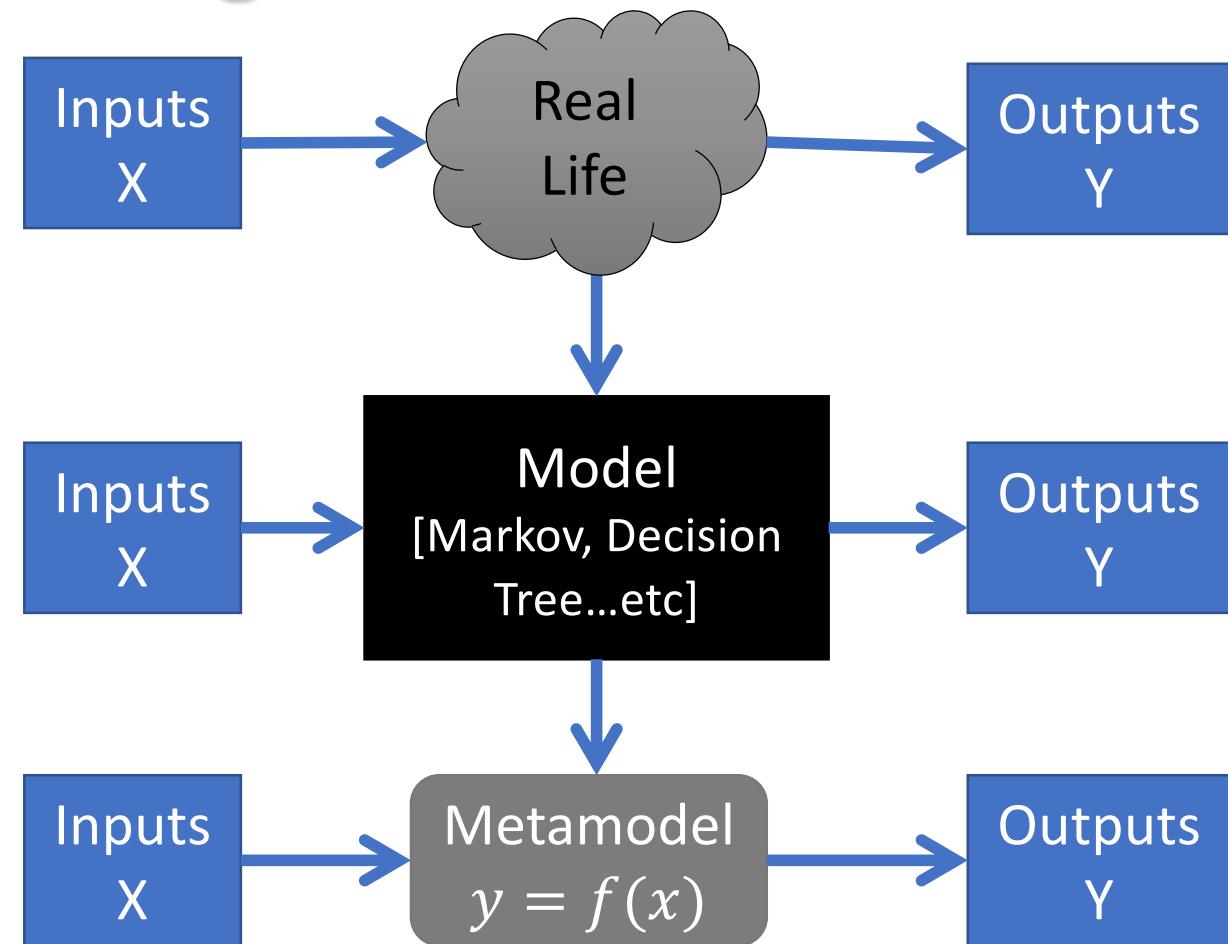
# Metamodeling



# Metamodeling



# Metamodeling



# History

- Widely used in computer science & engineering
  - Surrogates of simulation models
  - Half a century
  - Faster and cheaper replacements of simulation models
- Recent growth in its application in health policy and medical decision making

# Why regression works better on models than observational data

- In simulation models
  - Y are truly dependent
  - X are truly independent
  - No reverse causality
- Sample size is not an issue
- No missing data
- No omitted variables
- No measurement error
- Most models in MDM can be approximated as a linear metamodel

# Metamodel Types

- Regression analysis
  - Linear
    - Most common
    - Simple
    - Desirable properties
  - Nonlinear
    - Generalized additive model
- Gaussian processes
- Artificial Neural Networks

# The Metamodel in Simulation Analysis: Can It Be Trusted?

LINDA WEISER FRIEDMAN and ISRAEL PRESSMAN

Department of Statistics and Computer Information Systems,  
Baruch College of the City University of New York, USA

The use of a mathematical metamodel such as a regression model, constructed from simulation data and used to aid in the analysis of the simulated system, has been studied in recent years. For practitioners, the vast benefits of establishing a functional relationship among the variables in an unfamiliar and complex simulated system may be largely overshadowed by the concern that the metamodel, being a strongly data-based technique, may be valid only for the one particular set of simulation-generated data that went into it, which is to say not valid at all. Based on a study of 30 simulation experiments using three different simulation models, the authors conclude that the simulation metamodel is a reliable and valid technique to use in post-simulation analysis, and is probably just as good as the simulation model on which it is based.

## CONCLUSION

The results of this study indicate that the simulation metamodel tends to be relatively stable. Simulation practitioners and researchers should not be afraid to use a metamodel analysis because of concern that a data-dependent technique will produce different results with a different set of data. In fact, results from all three systems showed that the metamodel analysis did not react to the random variation caused by different sets of random-number streams but was fairly consistent from one set of data to the next. As a technique, it is probably just as good as the simulation upon which it is based.

# Sensitivity analysis

# Linear Regression Metamodeling as a Tool to Summarize and Present Simulation Model Results

Hawre Jalal, MD, MSc, Bryan Dowd, PhD, François Sainfort, PhD,  
Karen M. Kuntz, ScD

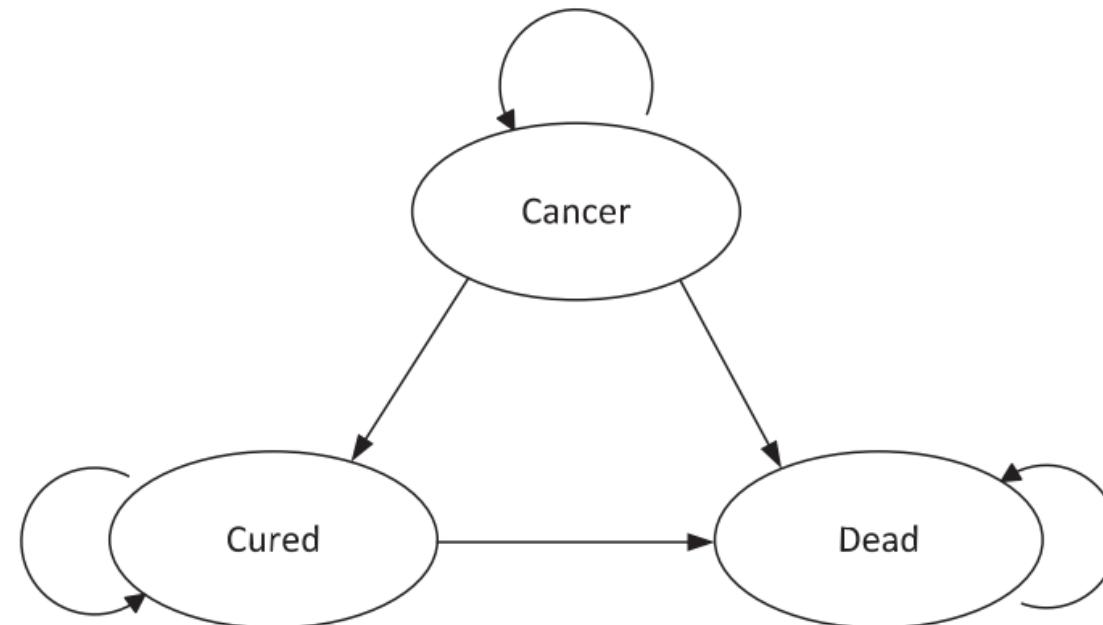
---

**Background/Objective.** Modelers lack a tool to systematically and clearly present complex model results, including those from sensitivity analyses. The objective was to propose linear regression metamodeling as a tool to increase transparency of decision analytic models and better communicate their results. **Methods.** We used a simplified cancer cure model to demonstrate our approach. The model computed the lifetime cost and benefit of 3 treatment options for cancer patients. We simulated 10,000 cohorts in a probabilistic sensitivity analysis (PSA) and regressed the model outcomes on the standardized input parameter values in a set of regression analyses. We used the regression coefficients to describe measures of sensitivity analyses, including threshold and parameter sensitivity analyses. We also compared the results of the PSA to deterministic

full-factorial and one-factor-at-a-time designs. **Results.** The regression intercept represented the estimated base-case outcome, and the other coefficients described the relative parameter uncertainty in the model. We defined simple relationships that compute the average and incremental net benefit of each intervention. Metamodeling produced outputs similar to traditional deterministic 1-way or 2-way sensitivity analyses but was more reliable since it used all parameter values. **Conclusions.** Linear regression metamodeling is a simple, yet powerful, tool that can assist modelers in communicating model characteristics and sensitivity analyses. **Key words:** sensitivity analysis; cost-effectiveness; economic evaluation; decision analysis; design of experiments; metamodel; regression. (*Med Decis Making* 2013;33:880–890)

# Example Model

- Markov model
- 50 yr cancer cohort
- 3 strategies
  - Chemotherapy
  - Radiation
  - Surgery
- 3 states
- Half-cycle correction
- Lifetime horizon

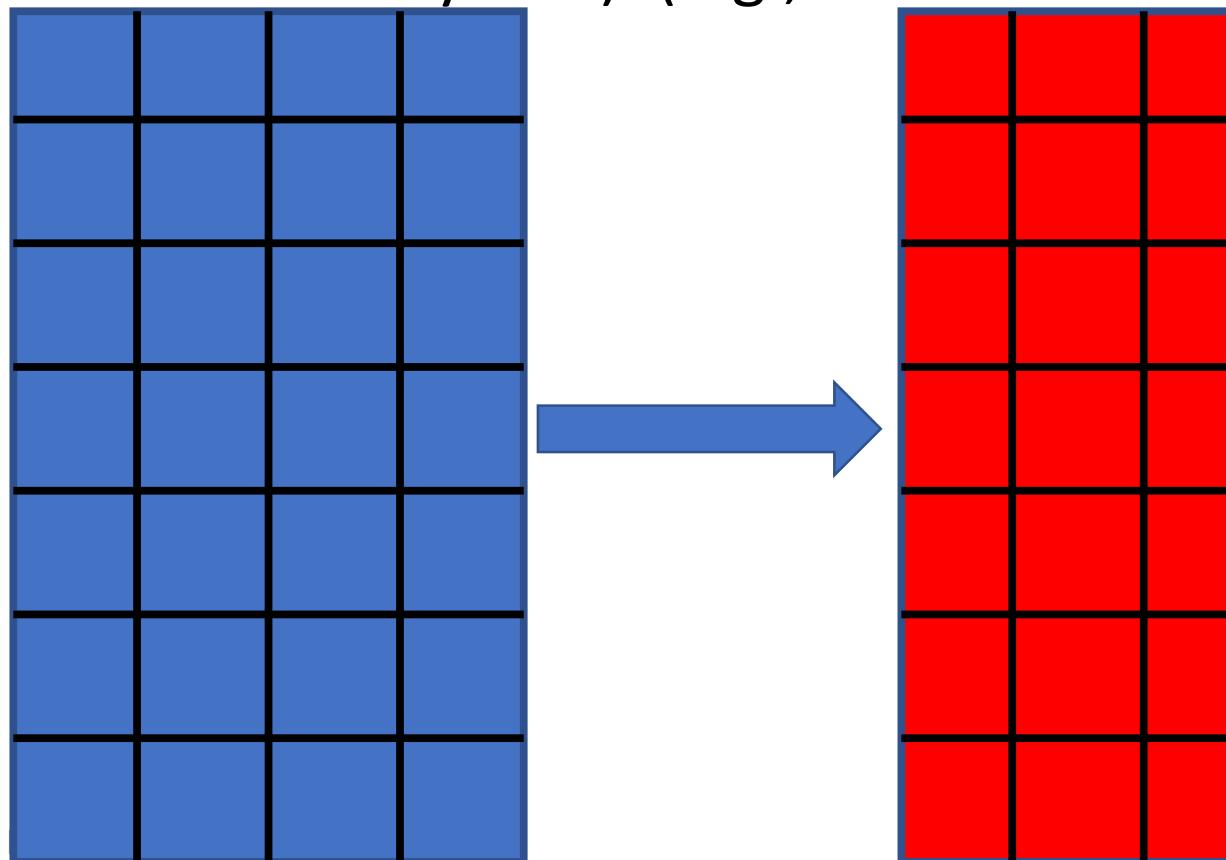


# Input Parameters

Parameter	Description	Distribution	Mean	s	LB	UB
pFailChemo	Annual probability of failing chemotherapy	Beta(45,55)	0.45	0.050	0.353	0.547
pFailRadio	Annual probability of failing radiotherapy	Beta(50,50)	0.5	0.050	0.402	0.598
pFailSurg	Probability of failing surgery	Beta(5,95)	0.05	0.022	0.007	0.093
pDieSurg	Probability of dying because of surgery	Beta(10,90)	0.1	0.030	0.041	0.159
$\mu_{\text{Cancer}}$	Cancer-specific mortality rate	LN(-1.69,0.4)	0.2	0.083	0.037	0.363
cChemo	Annual chemotherapy cost (\$)	LN(9.9,0.1)	20,000	2005	16,066	23,934
cRadio	Annual radiotherapy cost (\$)	LN(9.17,0.3)	10,000	3069	3979	16,021
cSurg	Surgery cost (\$)	LN(10.05,0.4)	25,000	10,414	4568	45,432

# PSA Dataset

Inputs ( $X$ )                          Outputs ( $Y$ )  
Independent Variables              Dependent Variables  
(e.g., cancer mortality rate)    (e.g., NHB Radiation Surgery)



# Linear Regression Metamodel

- Unstandardized

- $y = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + e$
- $\alpha$  = **marginal effect** of  $x$  on  $y$

- Standardized

- $z_j = \frac{x_j - \bar{x}_j}{\sigma_j}$
- $y = \beta_0 + \beta_1 z_1 + \beta_2 z_2 + e$
- $\beta$  = **marginal effect of one-standard deviation change in  $x$  on  $y$ .**
- $\beta_0 = \bar{y}$ : **base-case (mean) value of outcome**

$$\hat{\beta} = (X'X)^{-1}X'y$$

# Linear Regression Metamodel

## Unstandardized

	NHB
	Chemo
pFailChemo	-5.99*** (0.03)
pFailRadio	-0.004 (0.03)
pFailSurg	0.10 (0.07)
pDieSurg	-0.12** (0.05)
muDieCancer	-14.62*** (0.02)
cChemo	-0.000*** (0.000)
cRadio	0.000 (0.000)
cSurg	0.000 (0.000)
Constant	16.67*** (0.03)
Observations	10,000
R2	0.98

Note:  $p<0.1$ ;  $p<0.05$ ;  $p<0.01$   
NHB: Net health benefit

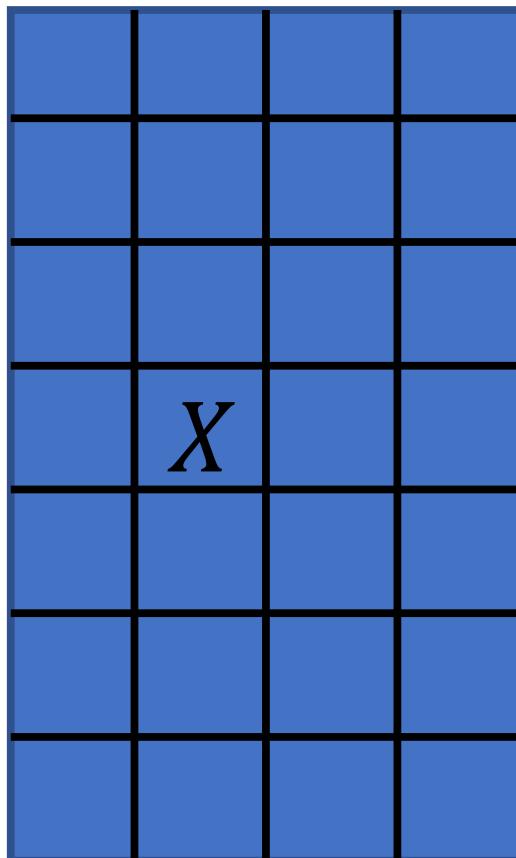
## Standardized

	NHB
	Chemo
pFailChemo	-0.30*** (0.002)
pFailRadio	-0.000 (0.002)
pFailSurg	0.002 (0.002)
pDieSurg	-0.003** (0.002)
muDieCancer	-1.20*** (0.002)
cChemo	-0.06*** (0.002)
cRadio	0.002 (0.002)
cSurg	0.002 (0.002)
Constant	10.45*** (0.002)
Observations	10,000
R2	0.98

Note:  $p<0.1$ ;  $p<0.05$ ;  $p<0.01$   
NHB: Net health benefit

# Prediction

$$X\hat{\beta} = \hat{y}$$



$$X \times \begin{matrix} \hat{\beta} \\ \vdots \end{matrix} = \begin{matrix} \hat{y} \\ \vdots \end{matrix}$$

The diagram illustrates the matrix multiplication  $X\hat{\beta} = \hat{y}$ . It shows a multiplication operation between a matrix  $X$  and a vector  $\hat{\beta}$ , resulting in a vector  $\hat{y}$ . The vector  $\hat{\beta}$  is represented as a column of four yellow squares with the symbol  $\hat{\beta}$  in the middle. The vector  $\hat{y}$  is represented as a column of five green squares with the symbol  $\hat{y}$  at the bottom.

# Threshold Analysis

- If  $y$  = incremental net health benefit
  - $\text{INHB} = \text{NHB}_1 - \text{NHB}_2$
- Then the threshold value at which the two compared outcomes are indifferent is at  $z_j^*$
- $\text{INB} = \theta_0 + \theta_1 z_1 + \theta_2 z_2 + u$
- $z_j^* = -\frac{\theta_0}{\theta_j}$

	$\theta_j > 0$	$\theta_j < 0$
$z_j > z_j^*$	Strategy 1	Strategy 2
$z_j < z_j^*$	Strategy 2	Strategy 1

# Multiple Multivariate Regression

- $Y$  = Matrix with multiple outcomes (dependent variables)
- $X$  = Matrix of input parameters
- $B$  = Matrix of coefficients
- Estimation of  $B$ :

$$\hat{B} = (X'X)^{-1}X'Y$$

# Prediction (multiple outcomes)

$$X \widehat{B} = \widehat{Y}$$

$x_1 \quad x_2 \quad \cdots \quad x_k$

$X$

$\times$

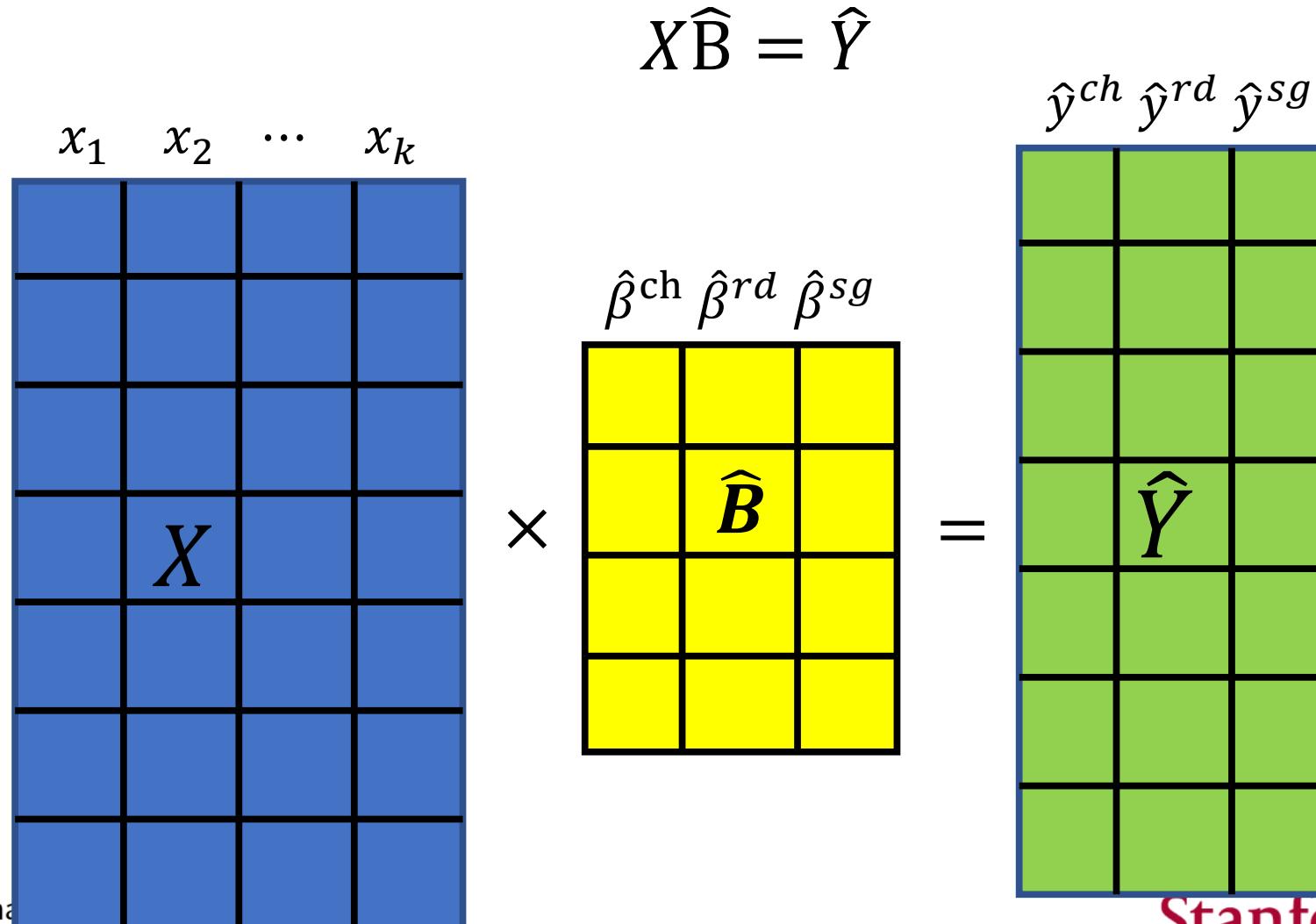
$\widehat{\beta}^{ch} \quad \widehat{\beta}^{rd} \quad \widehat{\beta}^{sg}$

$\widehat{B}$

$=$

$\widehat{y}^{ch} \quad \widehat{y}^{rd} \quad \widehat{y}^{sg}$

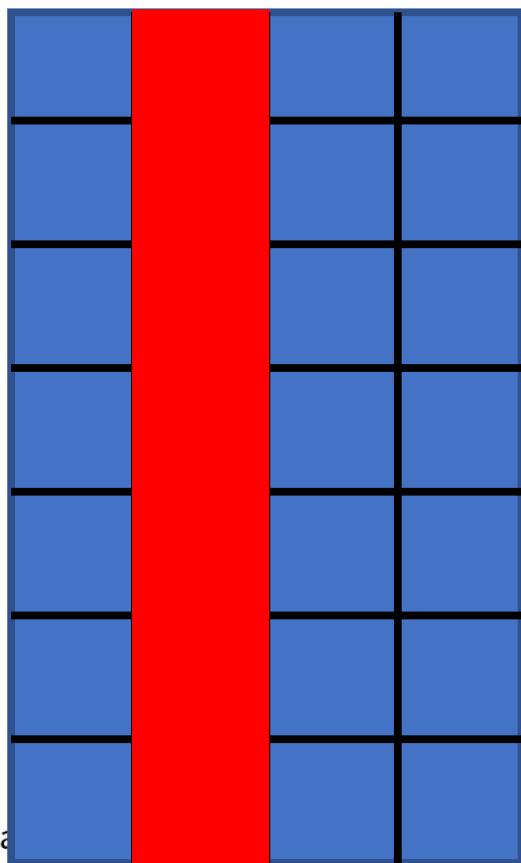
$\widehat{Y}$



# One-way sensitivity analysis (OWSA)

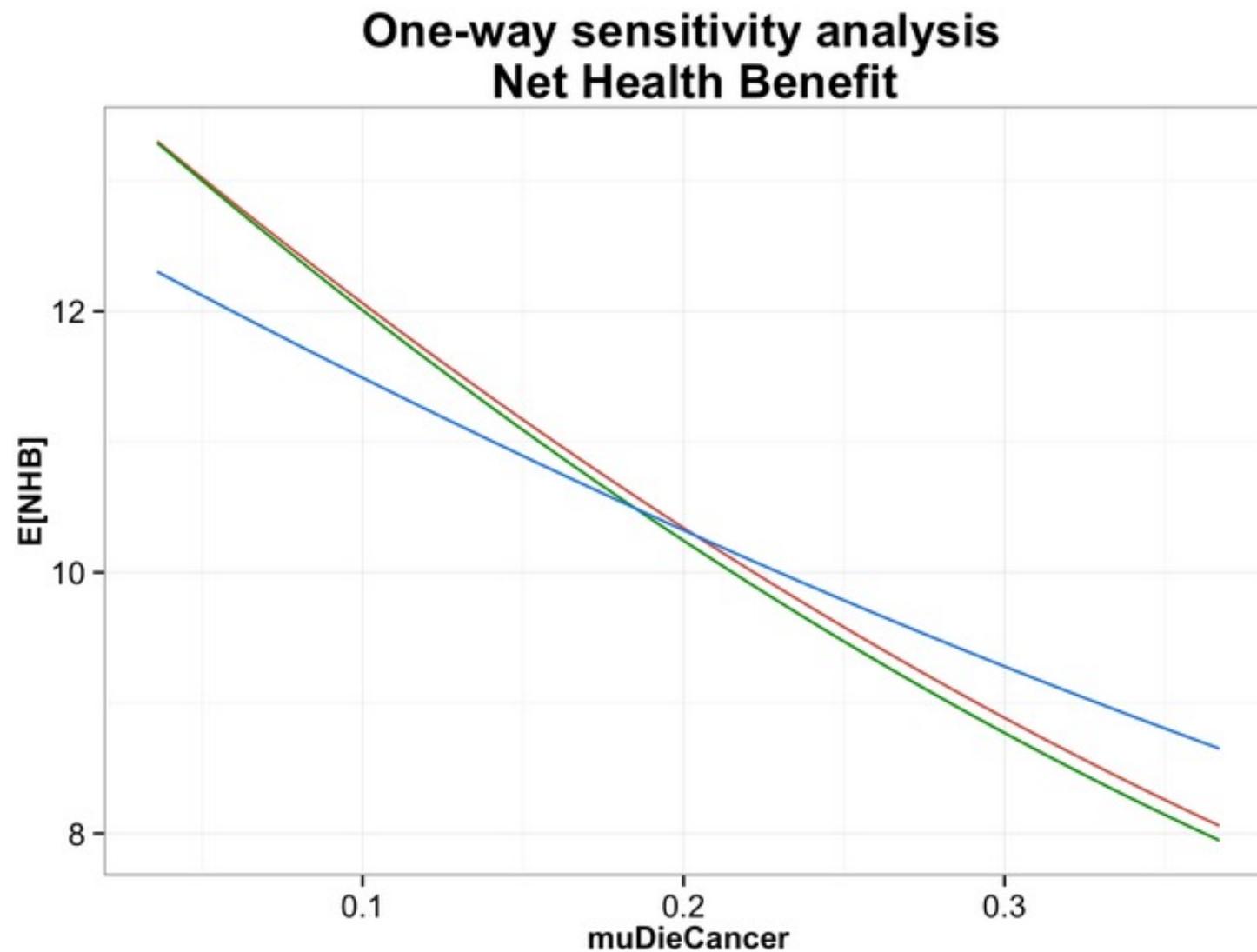
- Specify range for  $x_2$

- $\hat{Y} = X^* \hat{B}$      $\bar{x}_1 \quad x_2^* \quad \dots \quad \bar{x}_k$



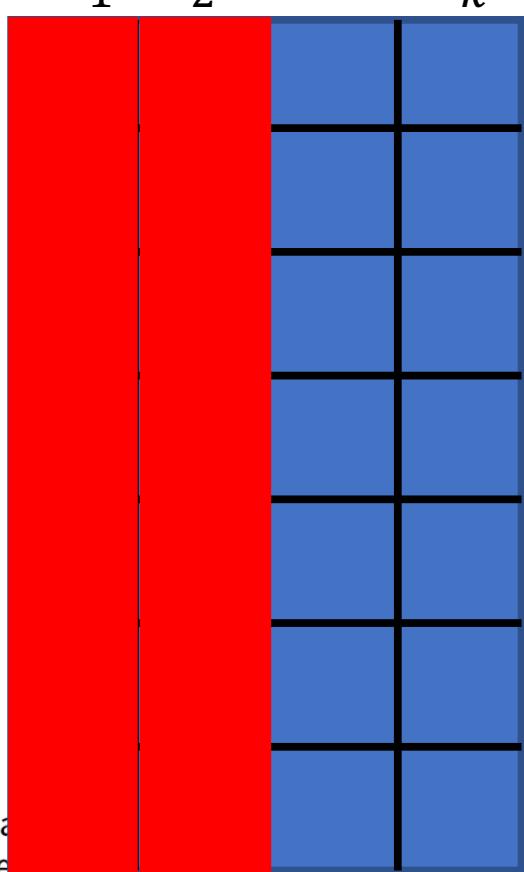
$$\begin{matrix} & \hat{\beta}^{ch} & \hat{\beta}^{rd} & \hat{\beta}^{sg} \\ \times & \begin{matrix} & & \\ & \hat{B} & \\ & & \end{matrix} & = & \begin{matrix} \hat{y}^{ch} & \hat{y}^{rd} & \hat{y}^{sg} \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \end{matrix} \end{matrix}$$

# One-way sensitivity analysis (OWSA)

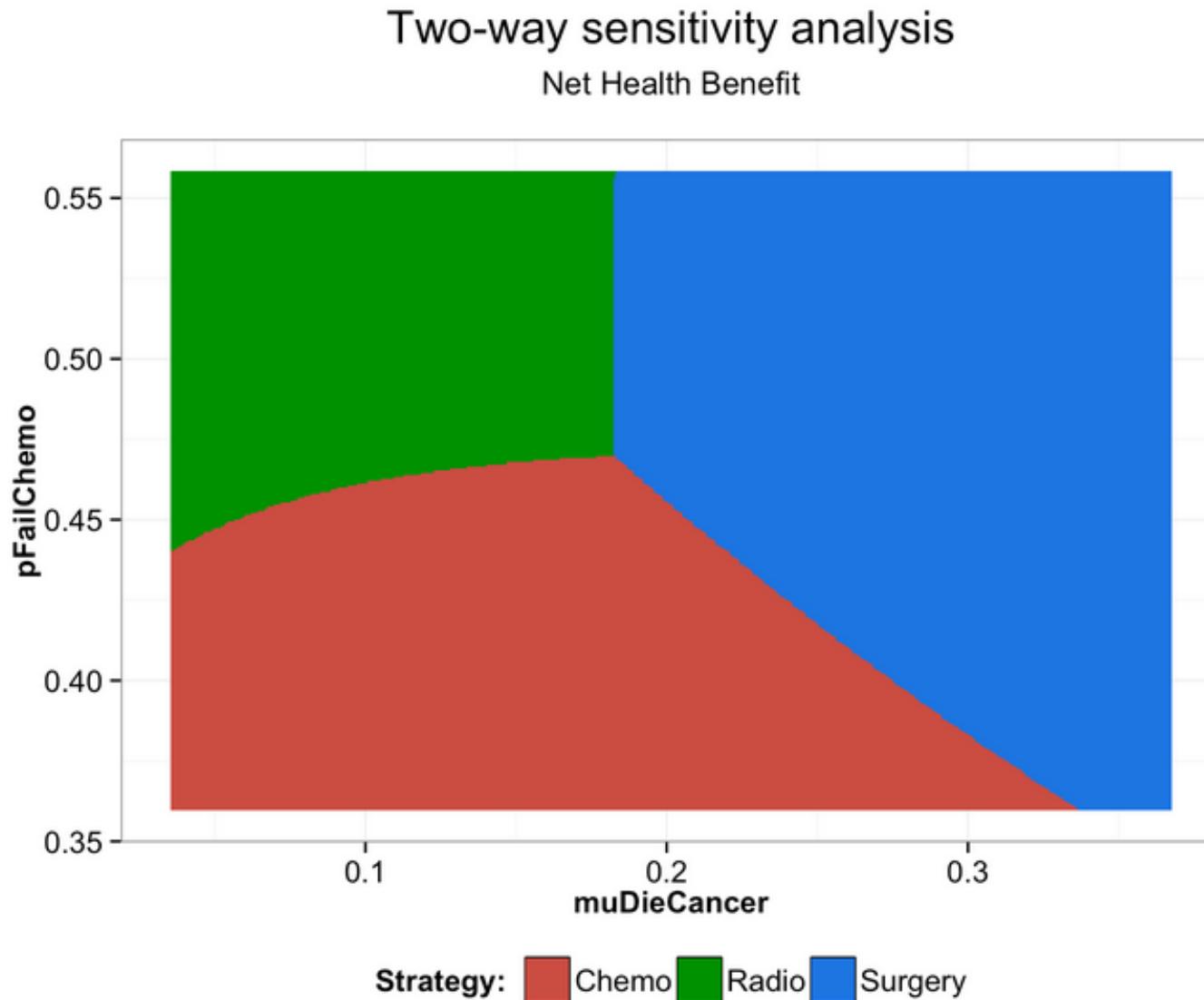


## Two-way sensitivity analysis (TWSA)

- Specify ranges for  $x_1$  and  $x_2$ 
  - Unique combinations
- $\hat{Y} = X^* \hat{B}$



# Two-way sensitivity analysis (TWSA)

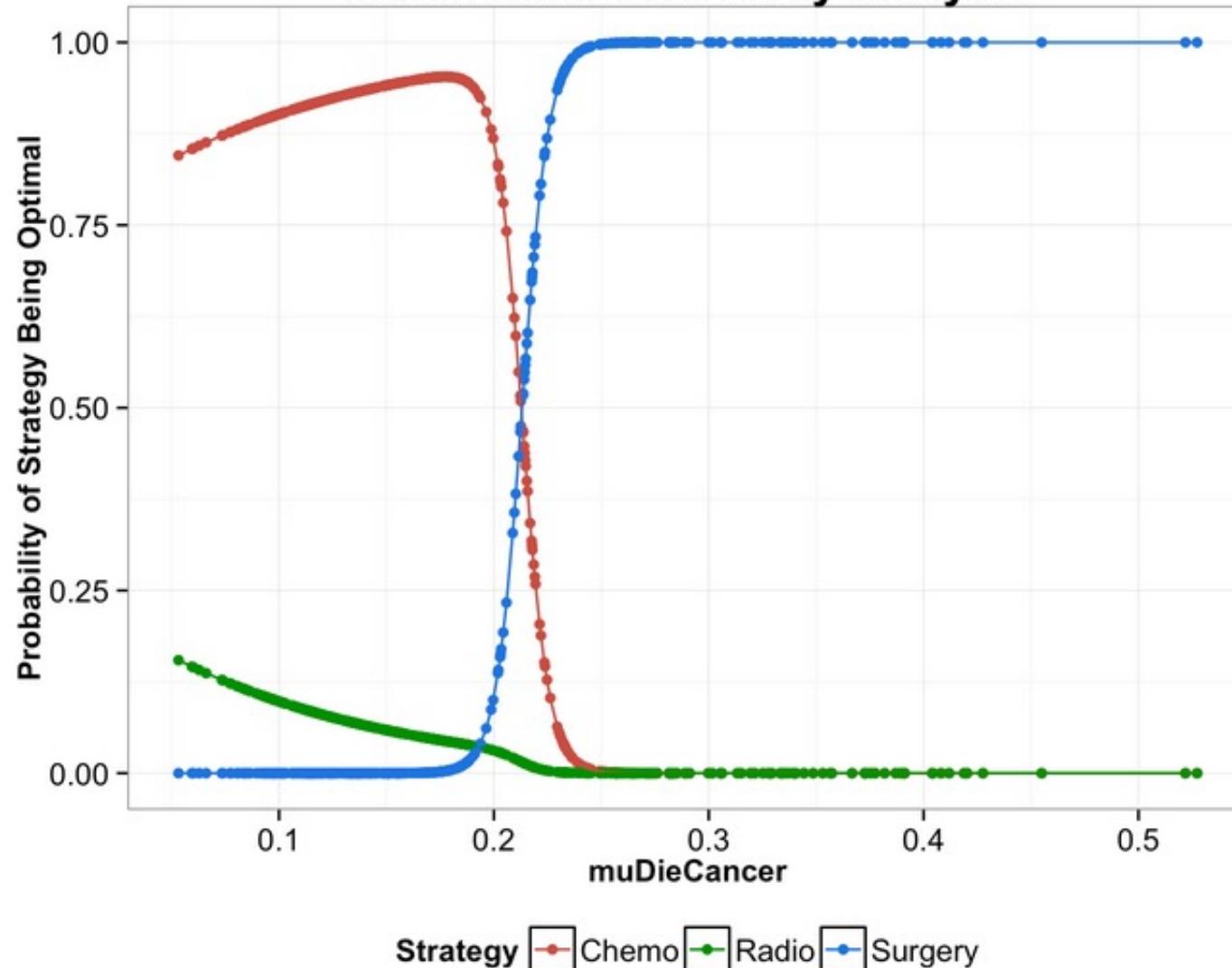


# Decision Sensitivity

- Fit a multinomial logistic regression
  - change in the probability of a strategy being the most cost-effective due to parameter uncertainty
$$Pr(\text{cost-effective intervention in simulation } i = j) \\ = \frac{\exp(\mathbf{z}'_i \boldsymbol{\beta}_j)}{\sum_{l=1}^J \exp(\mathbf{z}'_i \boldsymbol{\beta}_l)}$$
- $\boldsymbol{\beta}_j$  = change in the log odds of the  $j^{th}$  strategy being optimal relative to the base-case strategy being optimal due to a one-unit change in an input parameter's value

# Decision Sensitivity

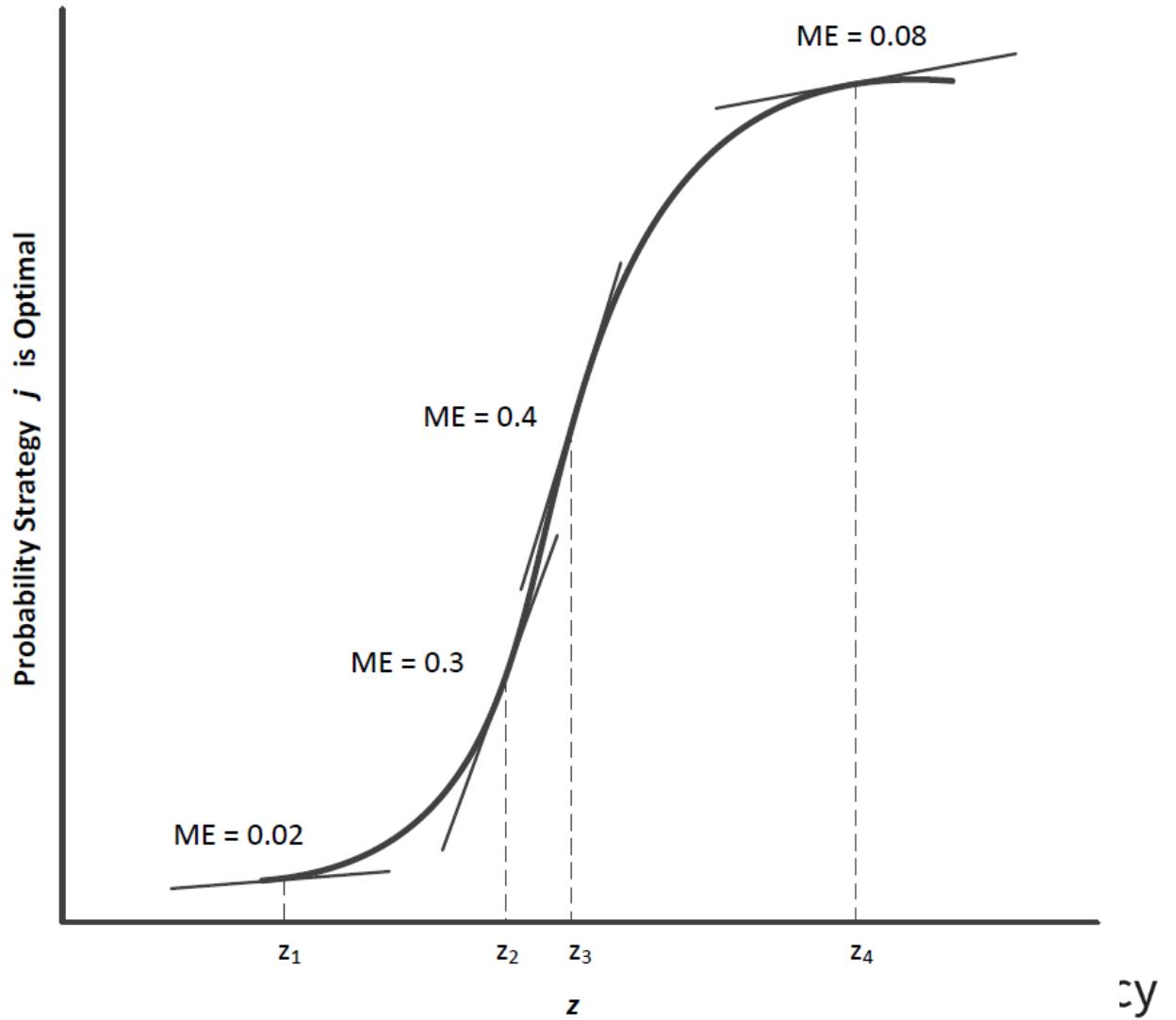
Multinomial sensitivity analysis



# Average Marginal Effects (AME)

$AME_j$  = Average change in the probability of the  $j^{\text{th}}$  strategy being cost-effective due to one unit change in an input parameter's value.

$$\text{AME}_j = \frac{1}{n} \sum_{i=1}^n p_{ij} \left( \beta_j - \sum_{l=1}^J p_{il} \beta_l \right)$$



# Extensions to the Regression Metamodel

- In general,

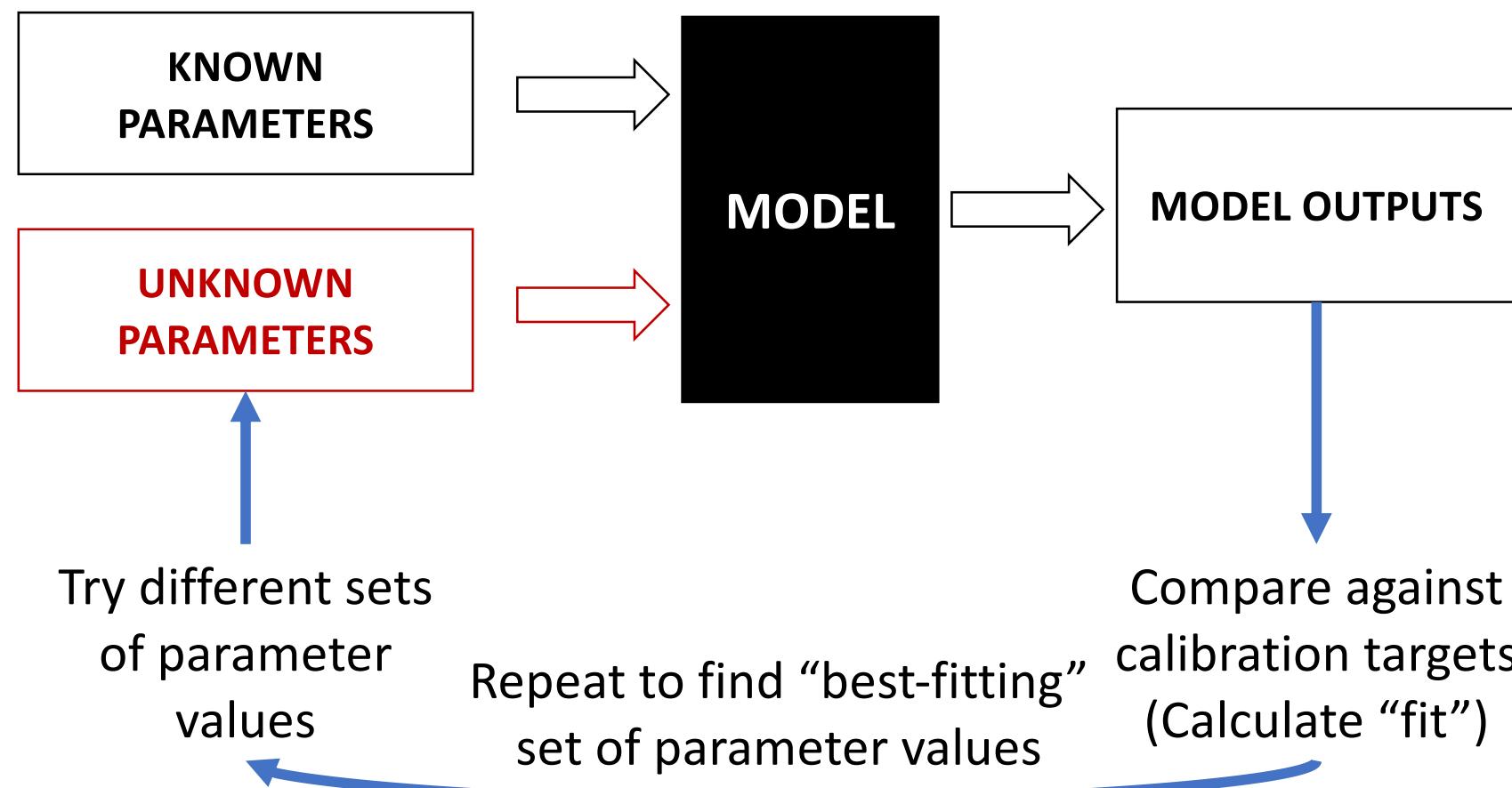
$$y = f(x_1, x_2, \dots, x_J; \theta) + e$$

where  $f(\quad)$  could be any linear or nonlinear function of the  $x$ 's.

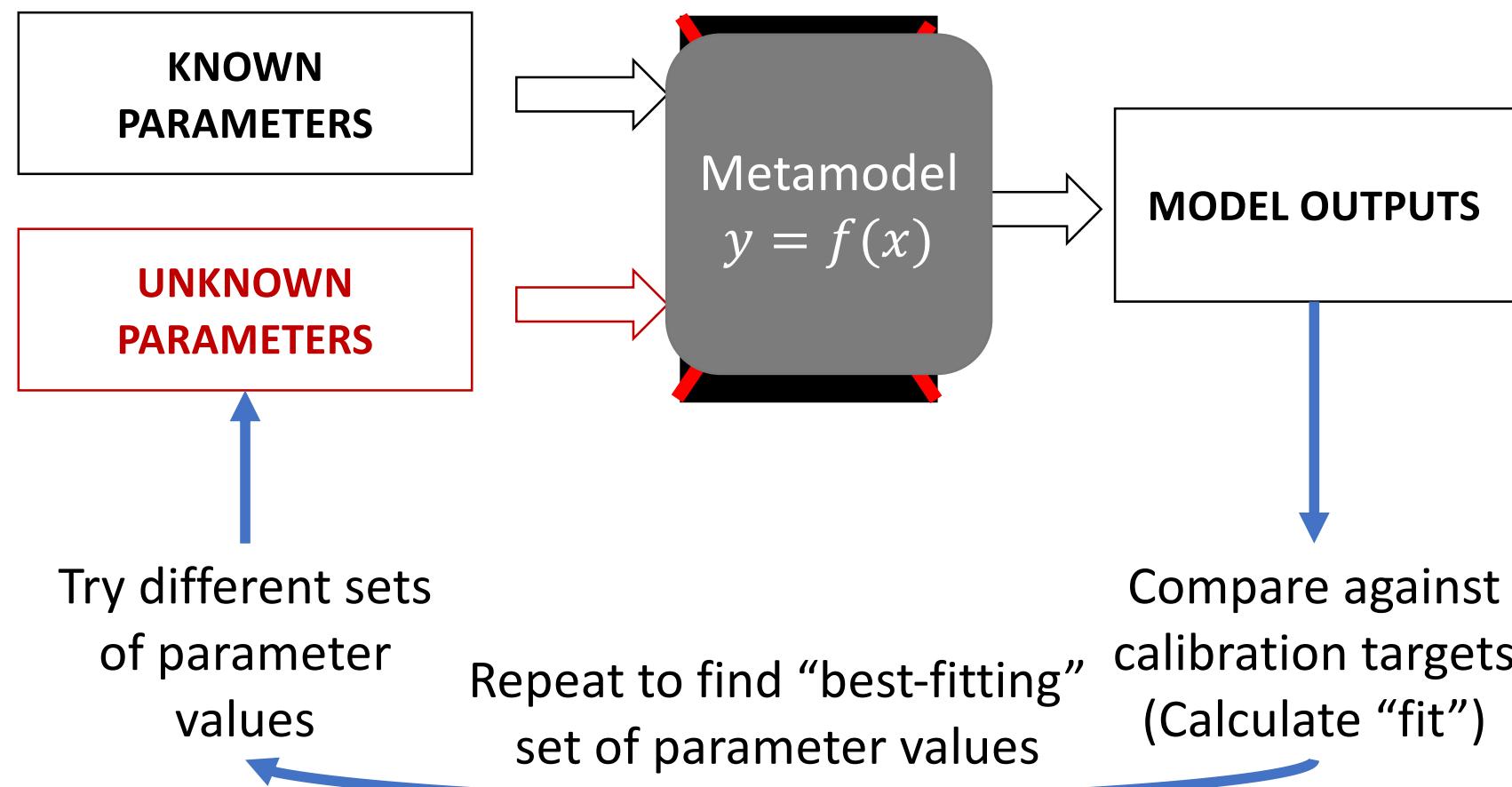
- For example, a polynomial, spline, generalized additive model (GAM), artificial neural network (ANN), etc.

# Emulator-Based Calibration

# Calibration process



# Calibration process with emulators

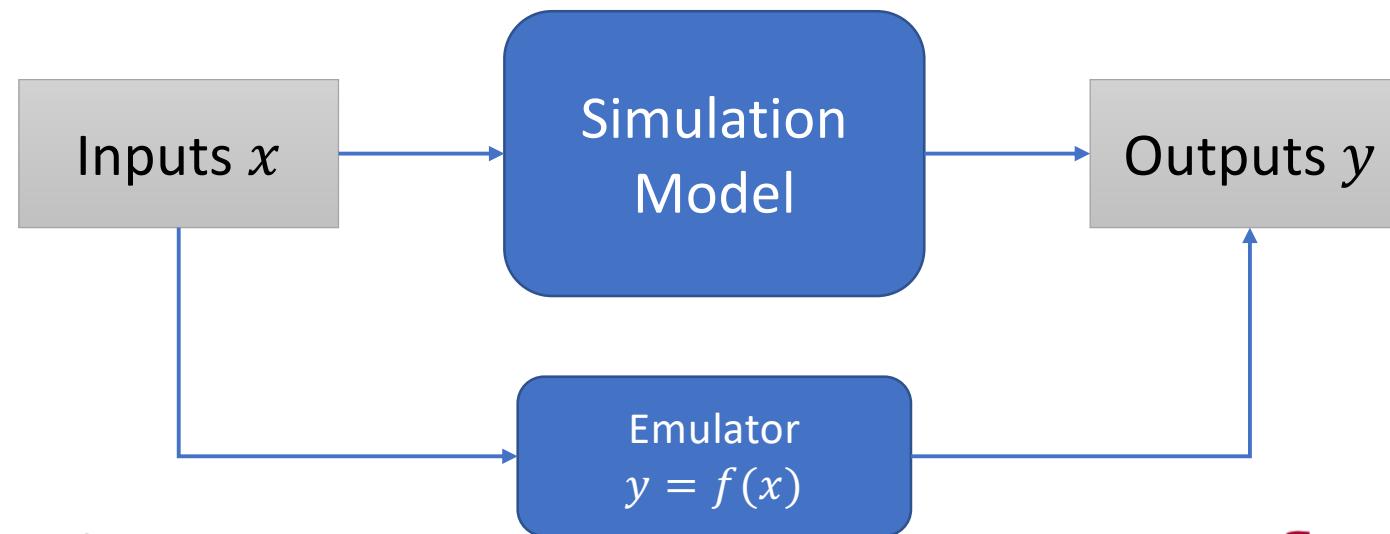


# Bayesian calibration

- Bayesian calibration
  - Quantification of uncertainty in model parameters
  - Produces joint posterior distributions of calibrated parameters
- However, it can be computationally intensive for moderately and highly complex models
- Development effort might require re-writing the model in a probabilistic language, such as BUGS or Stan

# Emulators = metamodels

- Emulators are statistical models that aim to replace the simulation model
- By mapping the relationship between inputs and outputs of simulation models



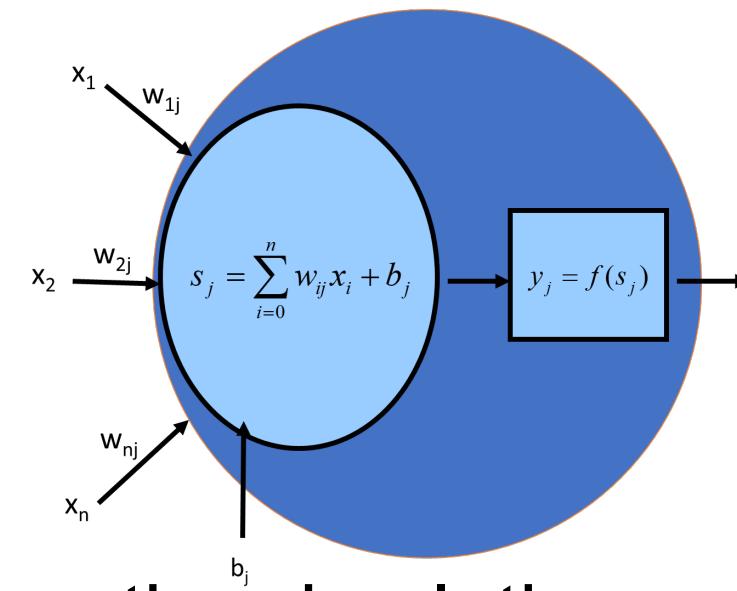
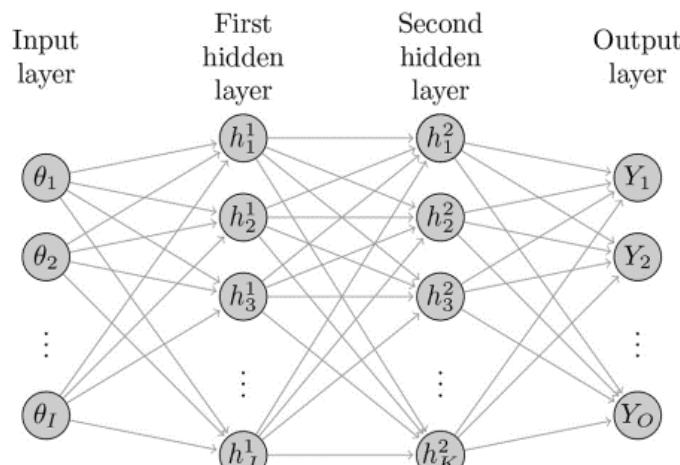
# Artificial Neural Networks (ANNs) as emulators

- ANNs can be excellent emulators because of their flexibility, strong predictive performance, and ease of coding within established probabilistic programs

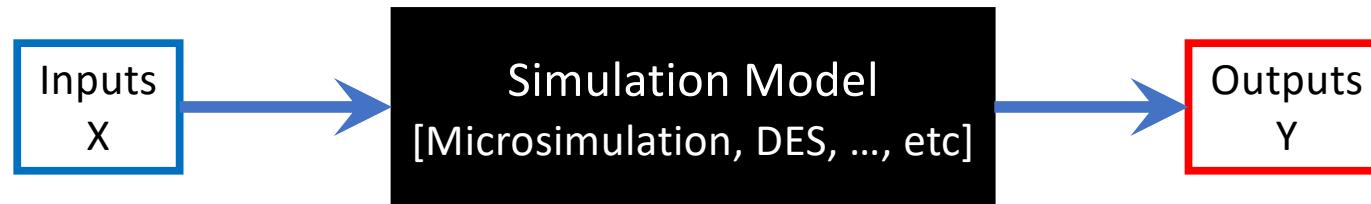


# Artificial Neural Networks (ANNs) as emulators

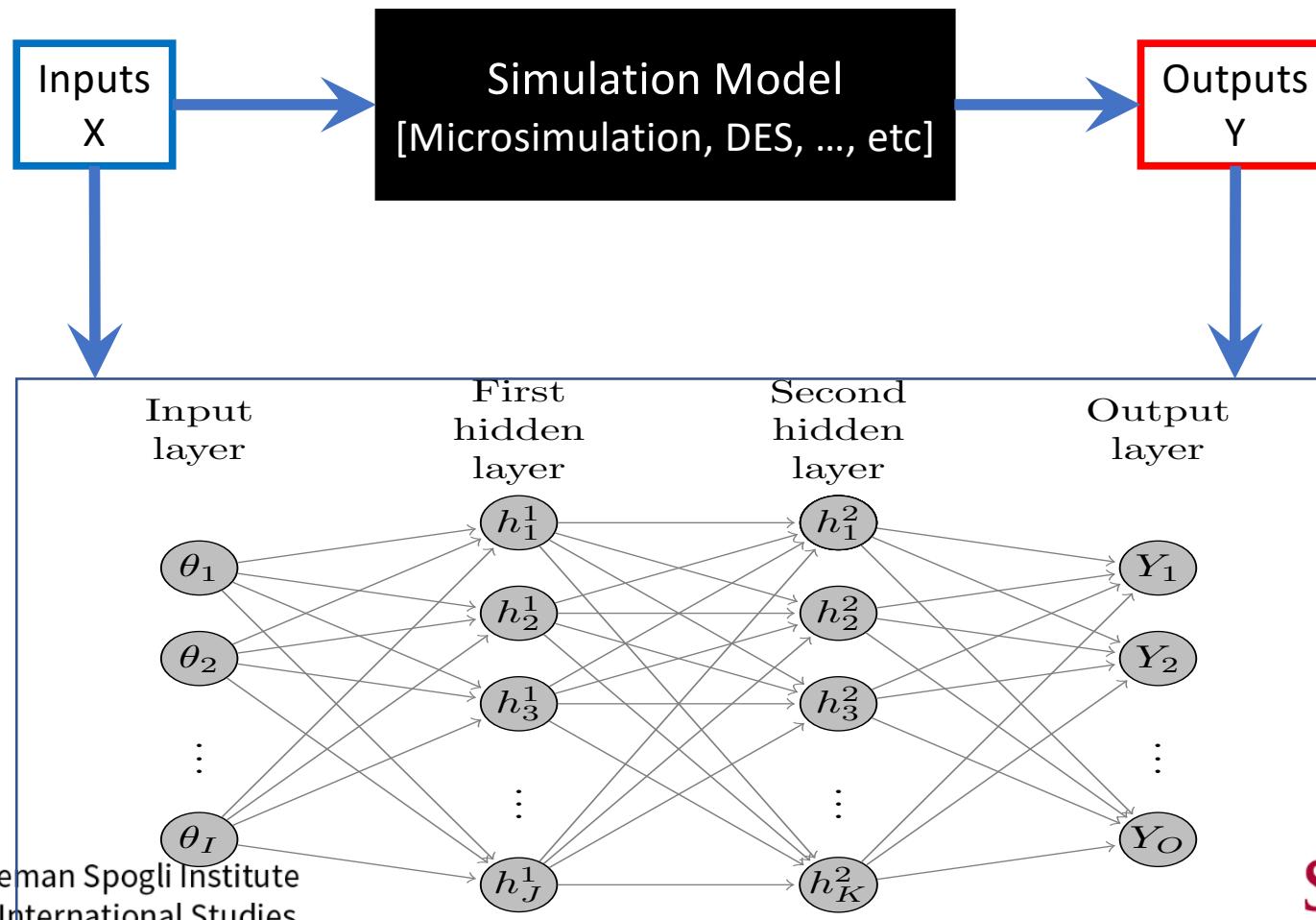
- ANNs can be excellent emulators because of their flexibility, strong predictive performance, and ease of coding within established probabilistic programs



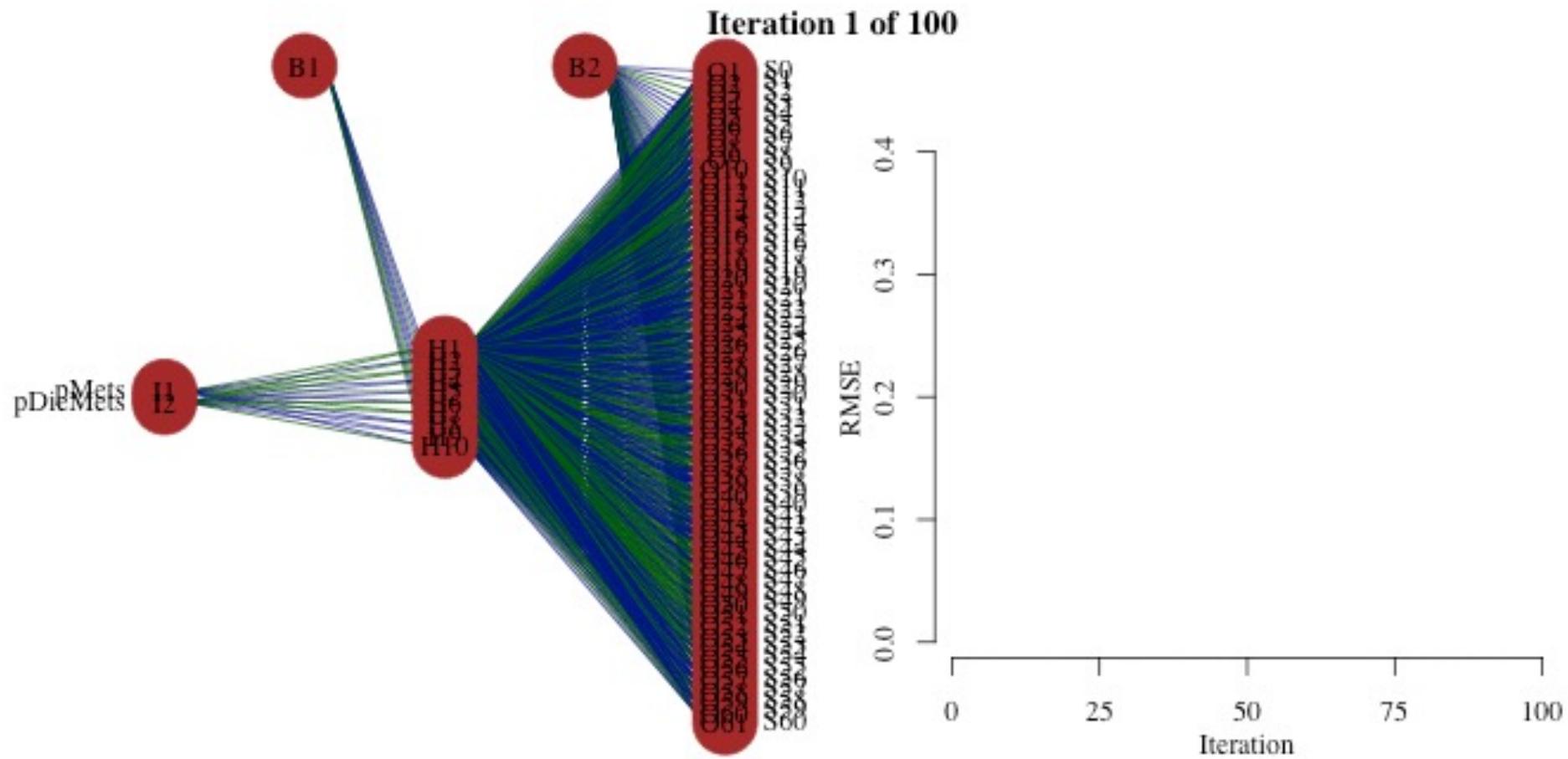
# ANN Metamodeling



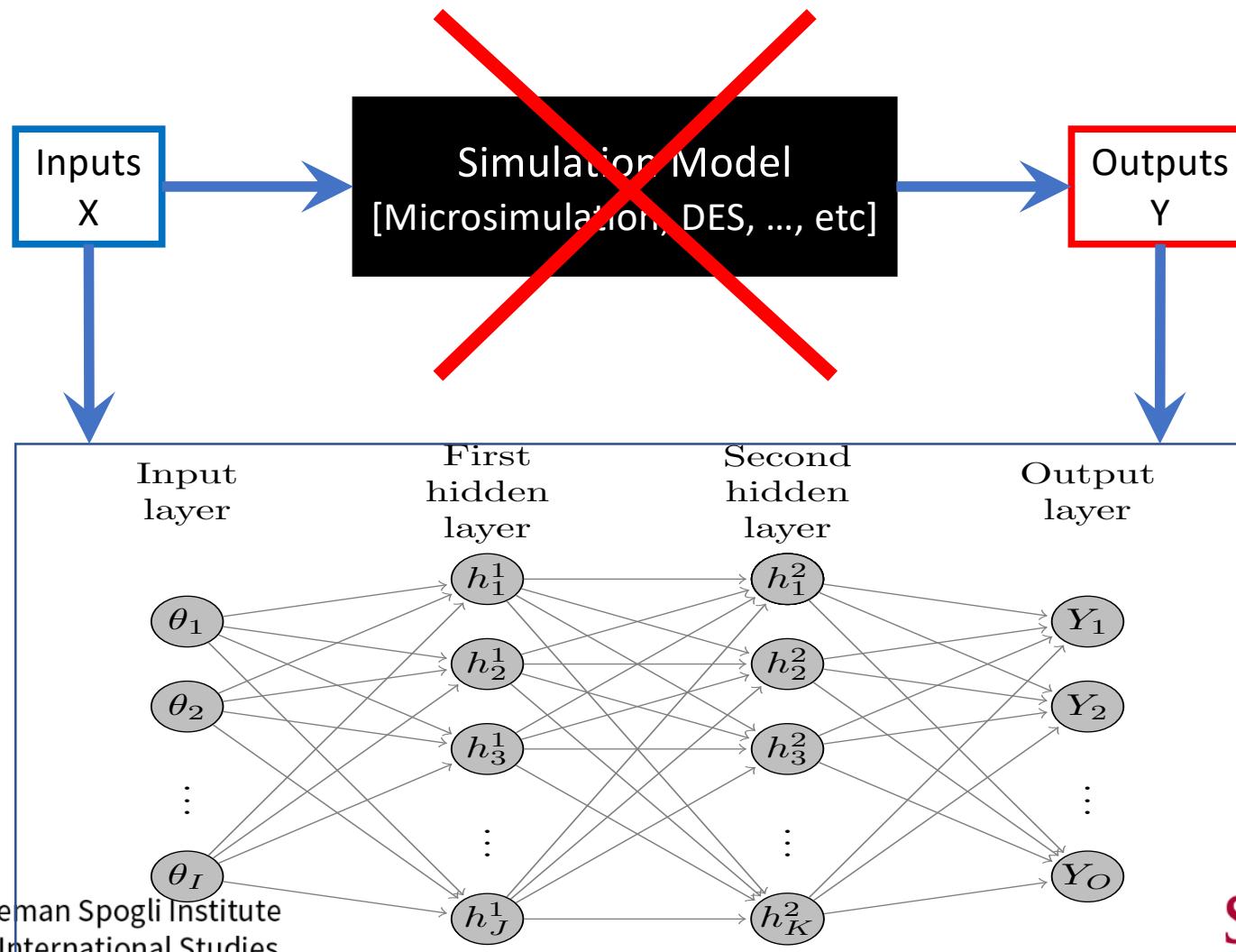
# ANN Metamodeling



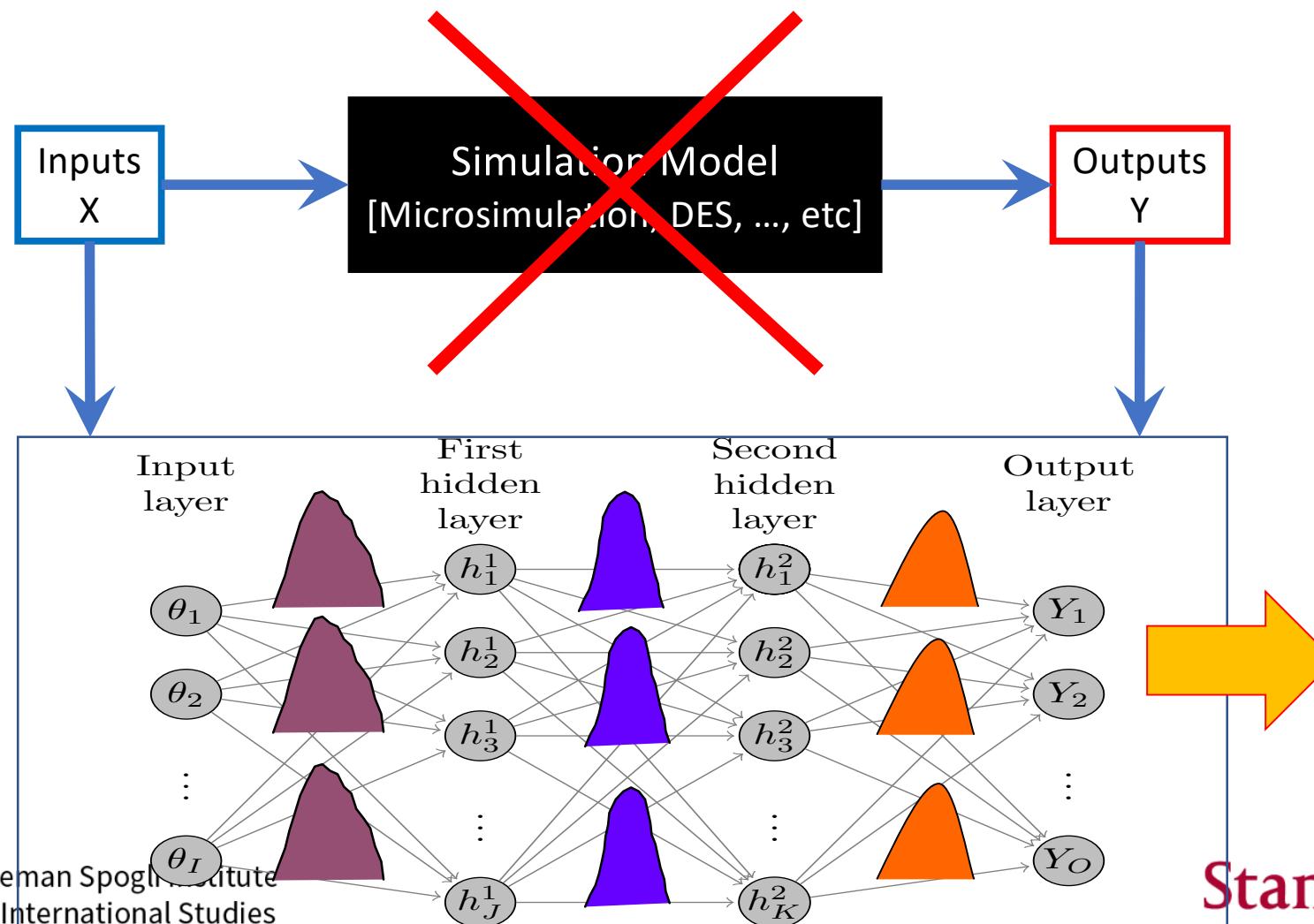
# Training the ANN



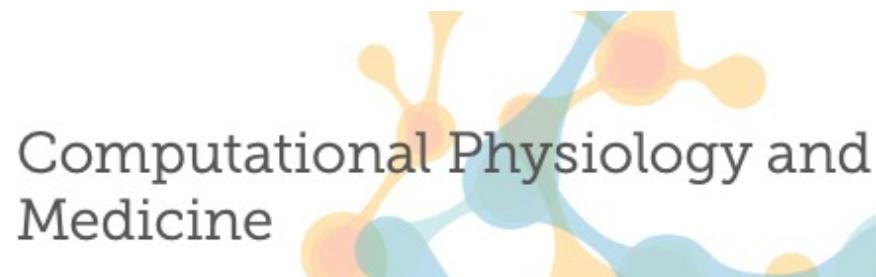
# ANN Metamodeling



# ANN Metamodeling for Bayesian Calibration



# Bayesian Calibration using Artificial Neural Networks (BayCANN)



METHODS article

Front. Physiol., 25 May 2021 | <https://doi.org/10.3389/fphys.2021.662314>



## BayCANN: Streamlining Bayesian Calibration With Artificial Neural Network Metamodeling

Hawre Jalal<sup>1\*</sup>, Thomas A. Trikalinos<sup>2</sup> and Fernando Alarid-Escudero<sup>3</sup>

Jalal, H., Trikalinos, T. A., & Alarid-Escudero, F. (2021). BayCANN: Streamlining Bayesian Calibration With Artificial Neural Network Metamodeling. *Frontiers in Physiology*, 12. <https://doi.org/10.3389/fphys.2021.662314>

**Stanford** Fréeman Spogli Institute for International Studies

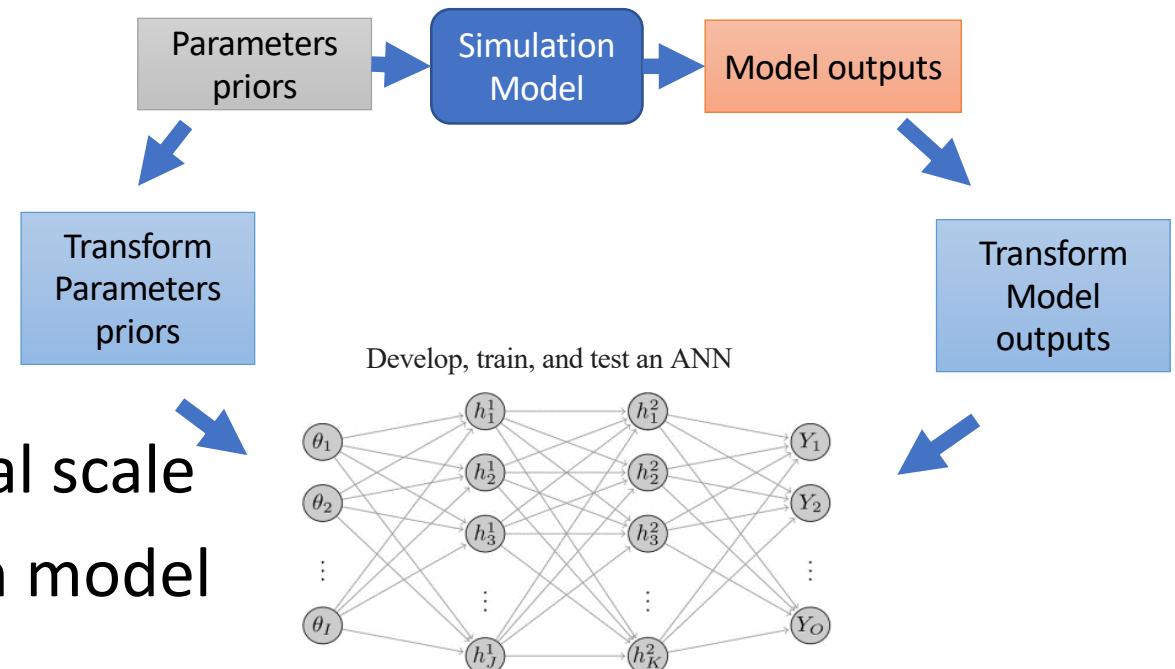
**Stanford** Health Policy

# Bayesian Calibration using Artificial Neural Networks (BayCANN)

- BayCANN is a practical approach to conducting Bayesian calibration in complex simulation models
- BayCANN involves:
  - Training an ANN emulator on a sample of model inputs and outputs
  - Calibrating the trained ANN emulator instead of the full model to obtain the posterior joint distribution

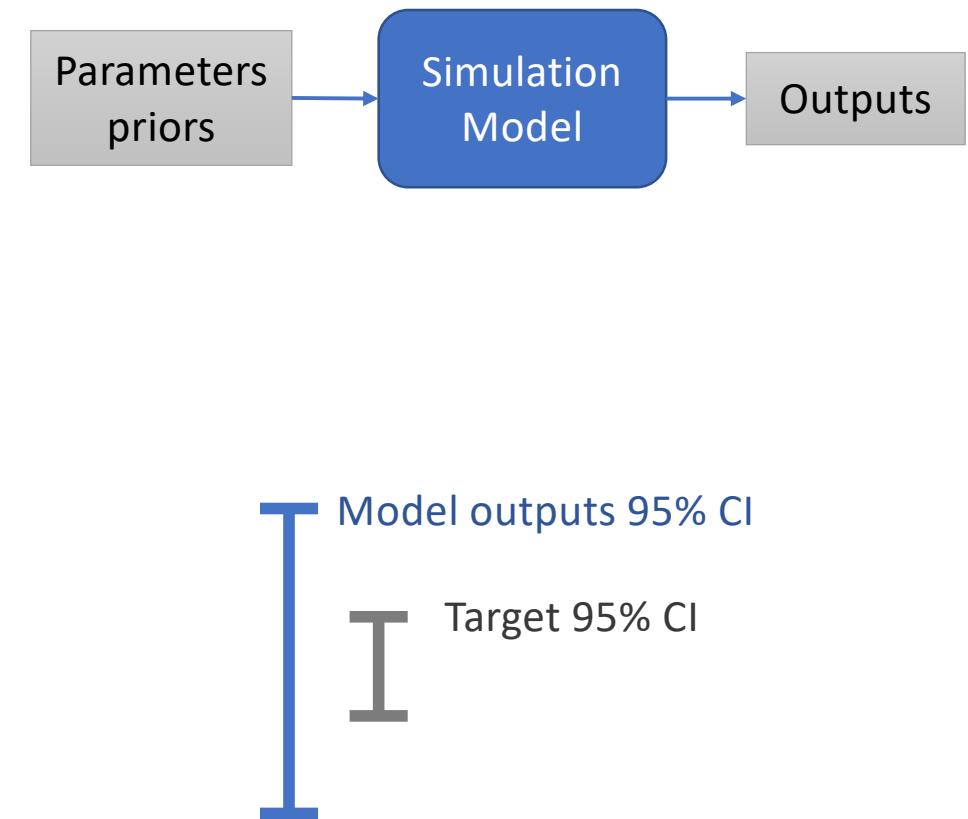
# BayCANN steps

1. Conduct a Latin hypercube sampling (LHS) from parameter priors
2. Generate model-predicted outputs for each parameter set from the LHS
3. Transform inputs and outputs
4. Train the ANN emulator
5. Code the ANN emulator in Stan
6. Estimate the posterior distribution
7. Transform inputs back to the original scale
8. Propagate uncertainty of simulation model
9. Validate



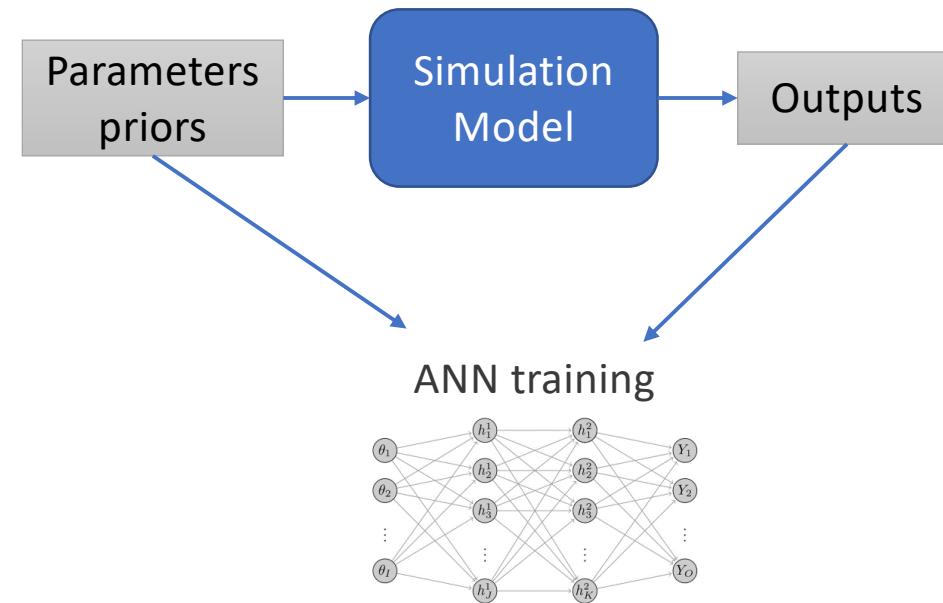
# Step 1: LHS

- Sampled parameter sets using a design of experiment (DoE) and generated corresponding outputs
- Verify that target uncertainty intervals are covered by model-predicted uncertainty



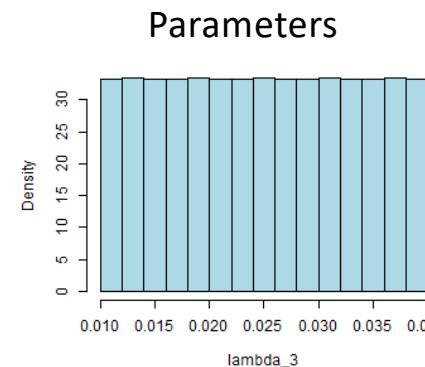
## Step 2: model predicted outputs

- Use samples to train an ANN for each model.



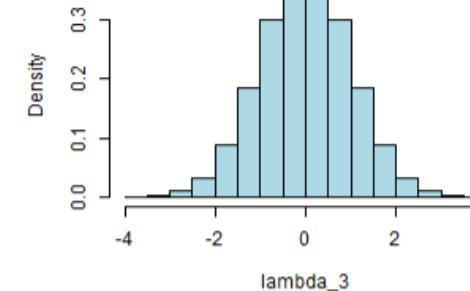
# Step 3: Transform Inputs and outputs

Original distributions



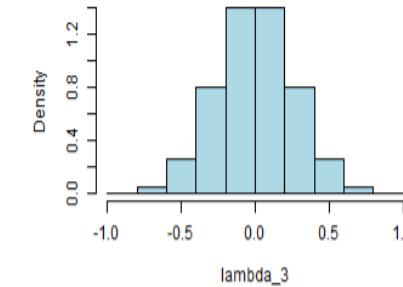
Normalization

$$f_p$$



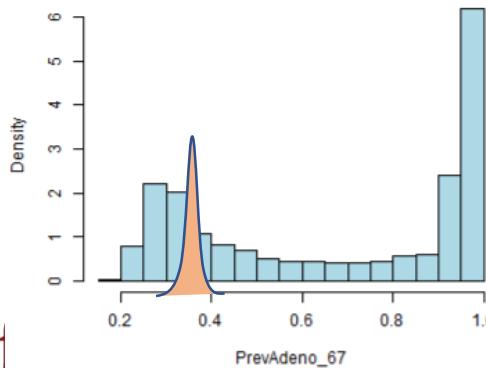
Scale

$$g_p$$

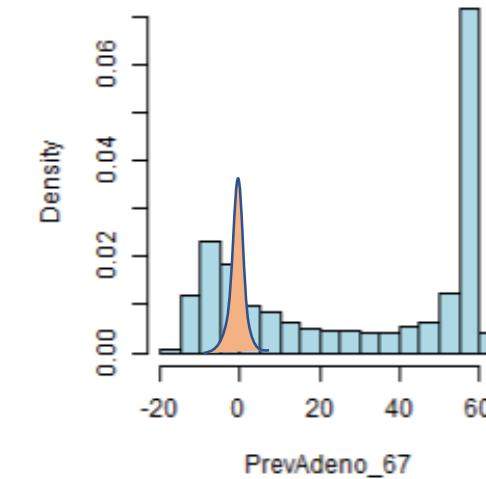


Output standardization

Model Outputs



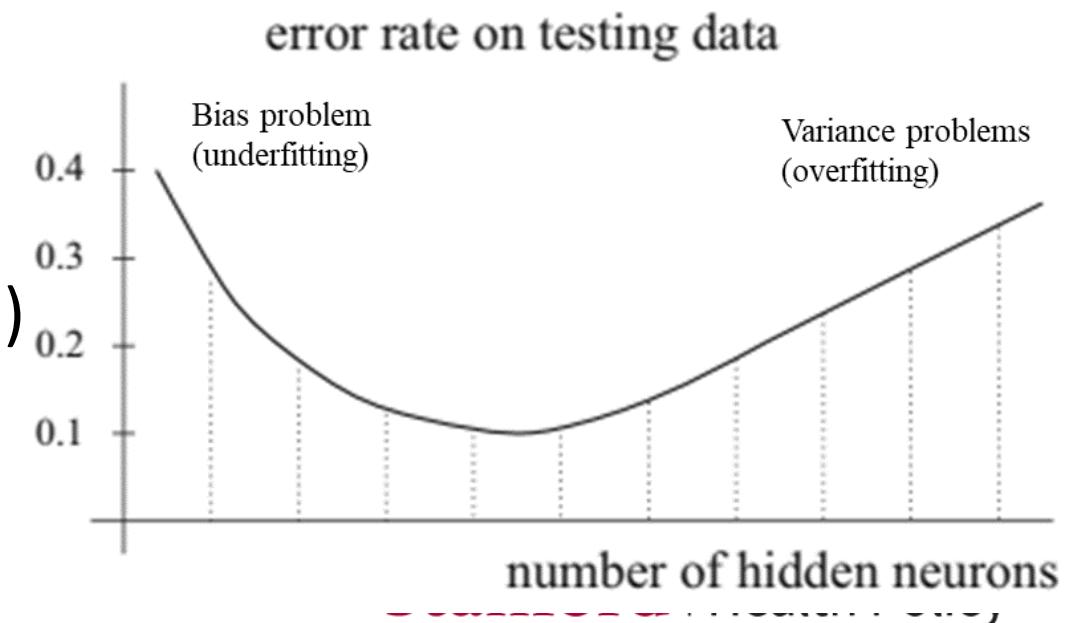
$$\frac{\text{output}_i - \text{target mean}_i}{\text{target sd}_i}$$



Stand

## Step 4: ANN training

- Elements required for training the ANN include the input data, the expected output and an objective function
- Identify the hyperparameters and ANN architectures
  - Hidden layers
  - Hidden nodes
  - Activation function
- By optimizing the objective function
  - Minimizing the mean square error (MSE)



# Step 4: ANN training

MSE by hidden nodes and hidden layers

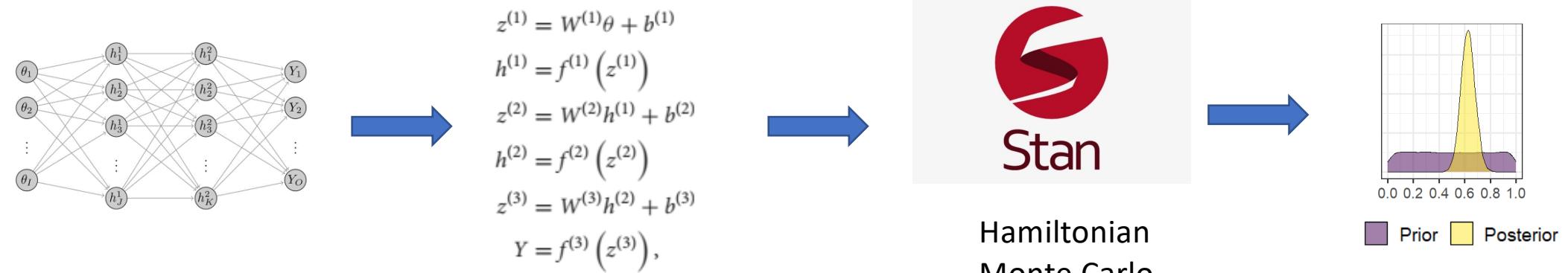


Sta

policy

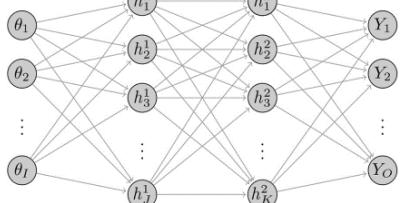
# Step 5 and 6: ANN in Stan

- We used the ANN with the lowest MSE as an emulator to conduct the Hamiltonian Monte Carlo-based Bayesian calibration.



# Step 7: Inverse transformation

Transform inputs and posteriors back to their original scale

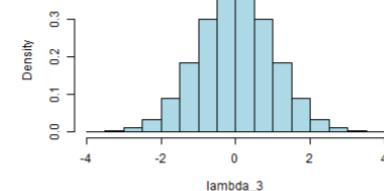


Hamiltonian  
Monte Carlo

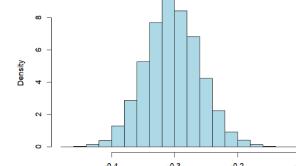
Unscale

$$g_p^{-1}$$

Parameters (prior)



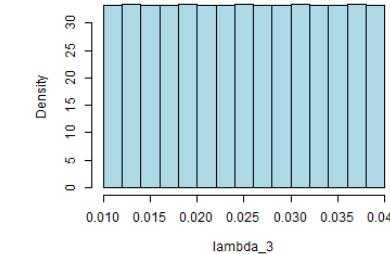
Parameters  
(Posterior)



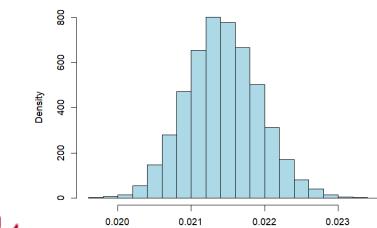
Back to original  
distribution

$$f_p^{-1}$$

Parameters (prior)

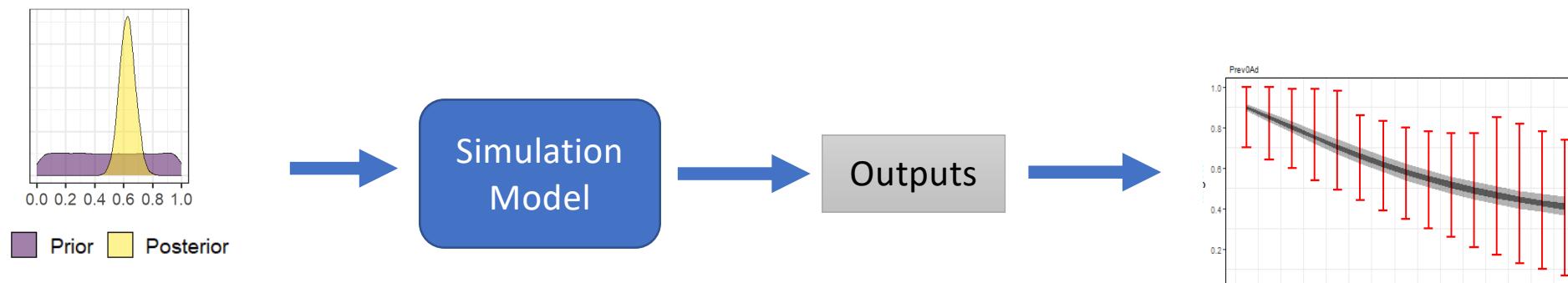


Parameter (Posterior)



# Steps 8 and 9: Propagate uncertainty and validate

- We compare the model-predicted outputs from the BayCANN joint posterior distribution against the calibration targets.



# Applied example

- Calibrate CISNET's SimCRC, MISCAN, and CRCSPIN natural history colorectal cancer (CRC) simulation models utilizing BayCANN.

*Original Research Article*



## Emulator-Based Bayesian Calibration of the CISNET Colorectal Cancer Models

*Medical Decision Making*  
2024, Vol. 44(5) 543–553  
© The Author(s) 2024  
Article reuse guidelines:  
[sagepub.com/journals-permissions](https://sagepub.com/journals-permissions)  
DOI: 10.1177/0272989X241255618  
[journals.sagepub.com/home/mdi](https://journals.sagepub.com/home/mdi)



**Carlos Pineda-Antunez**, **Claudia Seguin**, **Luuk A. van Duuren**,  
**Amy B. Knudsen**, **Barak Davidi**, **Pedro Nascimento de Lima**,  
**Carolyn Rutter**, **Karen M. Kuntz**, **Iris Lansdorp-Vogelaar**, **Nicholson Collier**,  
**Jonathan Ozik**, and **Fernando Alarid-Escudero**

<https://journals.sagepub.com/doi/full/10.1177/0272989X241255618>

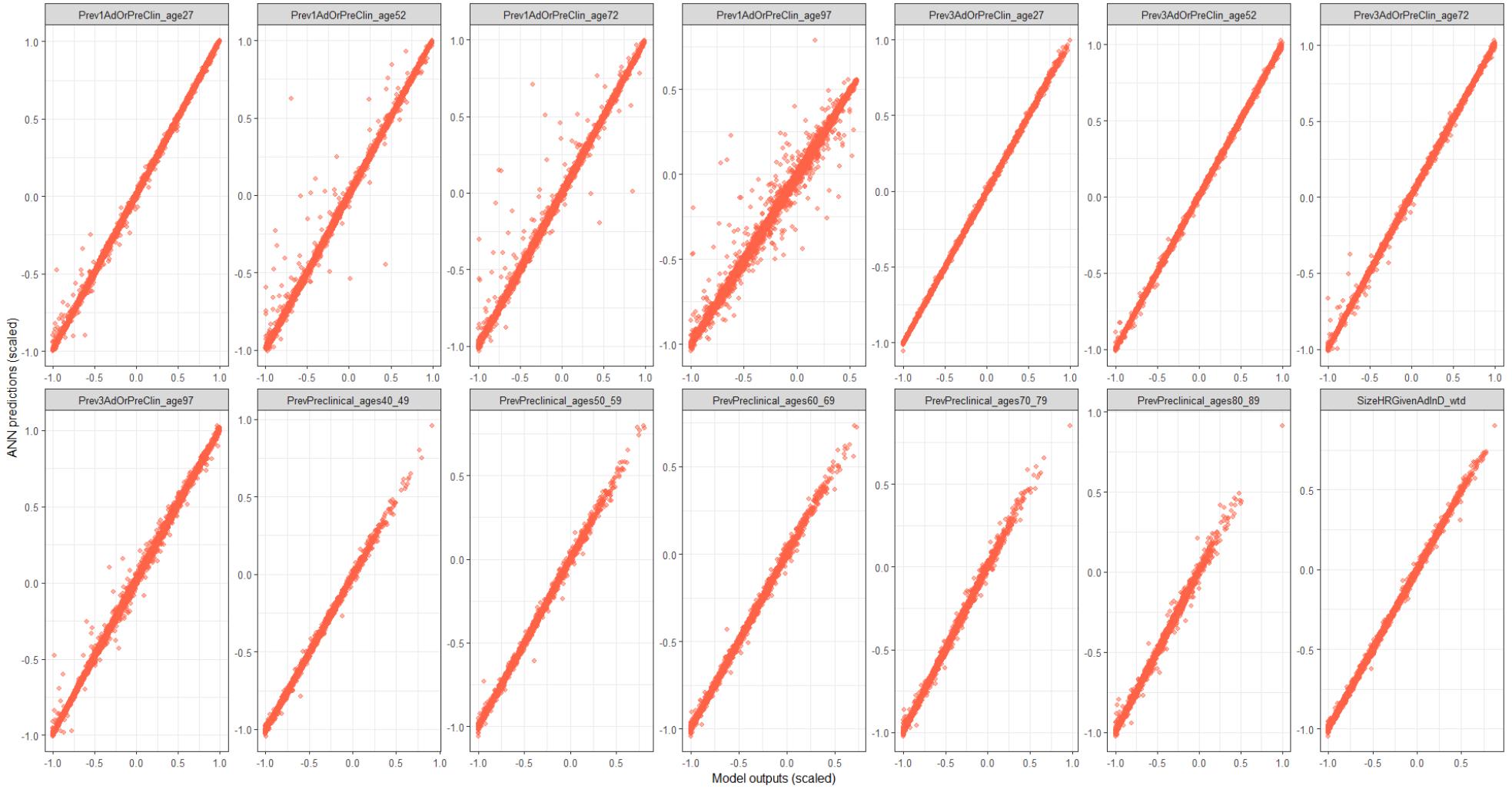
# Results: ANN structures

Selected ANN based on the lowest MSE

Model	Required transformation	Structure	Training time
<b>SimCRC</b>			
<i>30 parameters</i> <i>110 targets</i>	- Normalize inputs $\sim N(0,1)$ - Scale outputs (-1,1)	- 3 hidden layers - 450 hidden nodes - Activation function: tanh	39 min
<b>MISCAN</b>			
<i>37 parameters</i> <i>84 targets</i>	- Normalize inputs $\sim N(0,1)$ - Outputs standardized by targets	- 1 hidden layer - 600 hidden nodes - Activation function: tanh	23 min
<b>CRC SPIN</b>			
<i>22 parameters</i> <i>41 targets</i>	- Normalize inputs $\sim N(0,1)$ - Scale outputs (-1,1)	- 1 hidden layer - 140 hidden nodes - Activation function: sigmoid	10 min

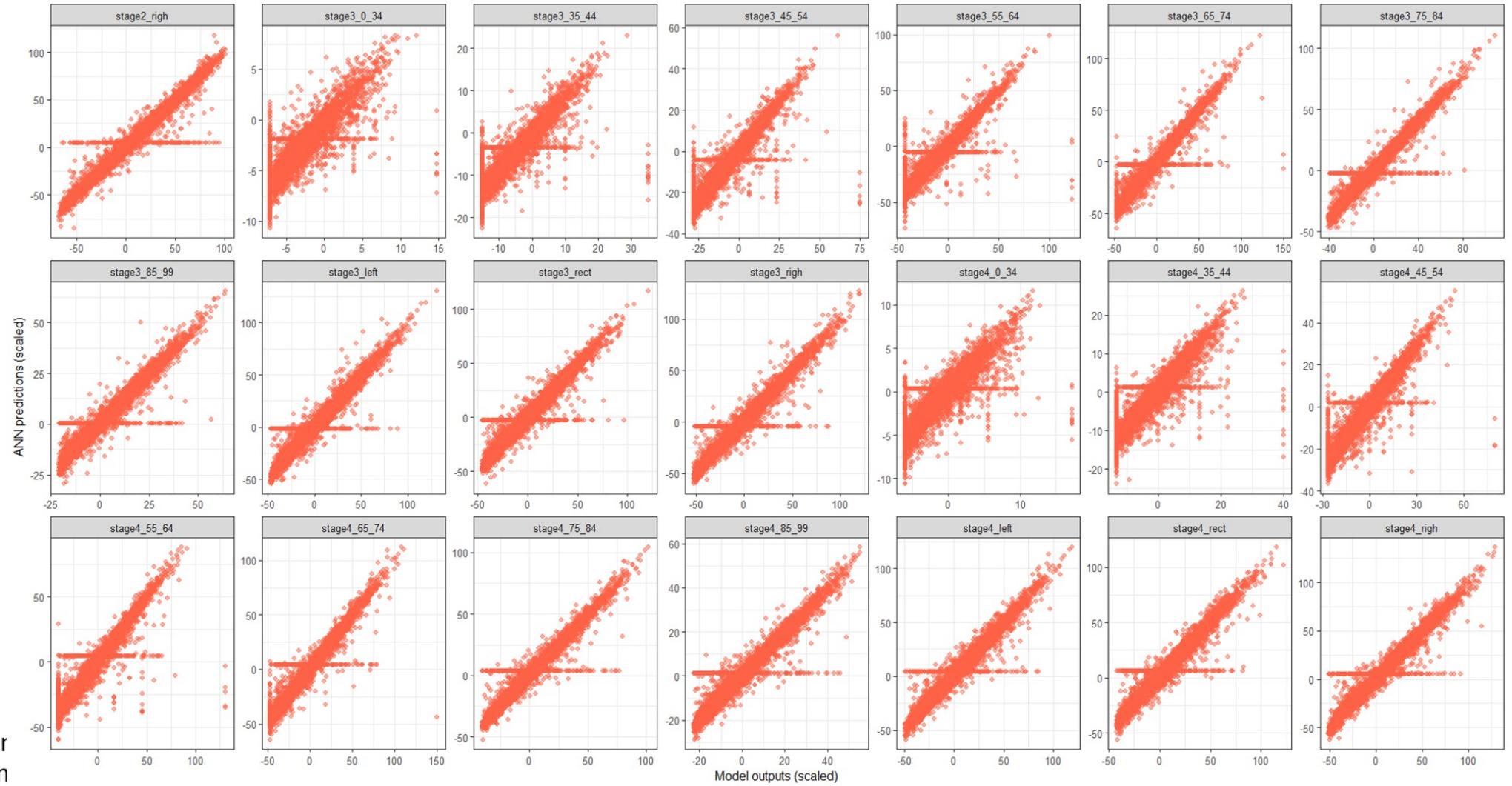
# Results: Prediction performance

SimCRC



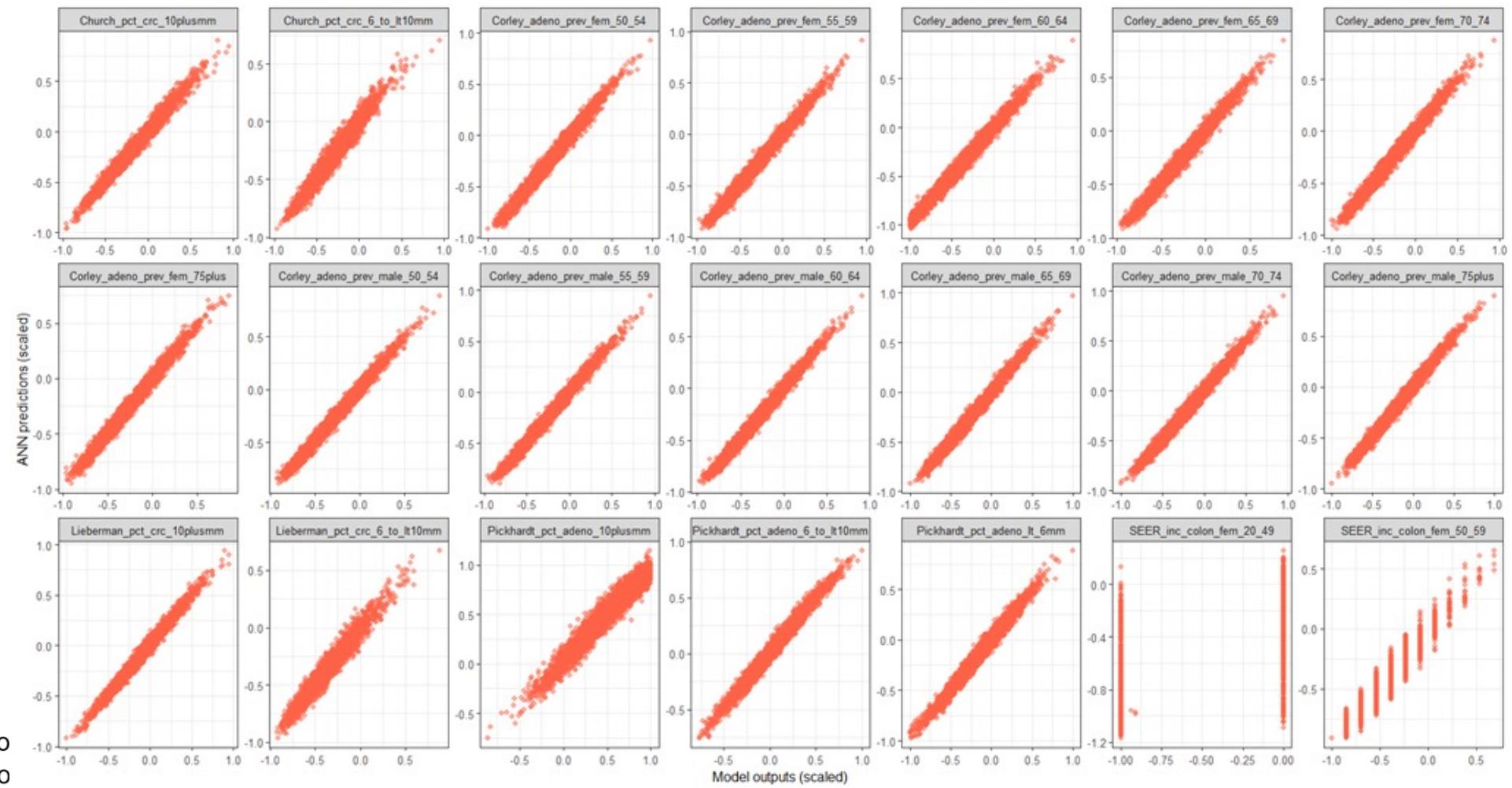
# Results: Prediction performance

MISCAN

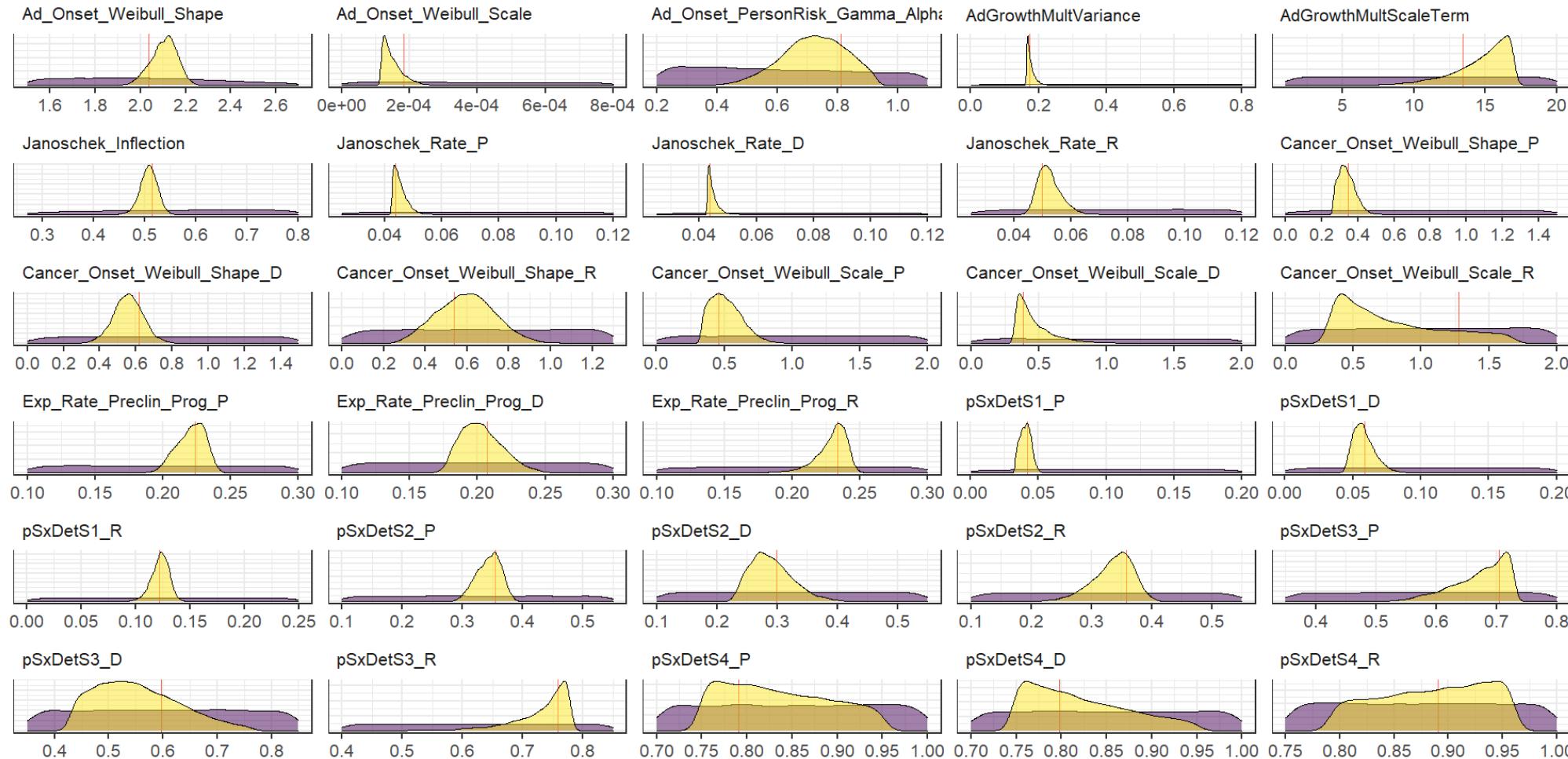


# Results: Prediction performance

CRC-SPIN

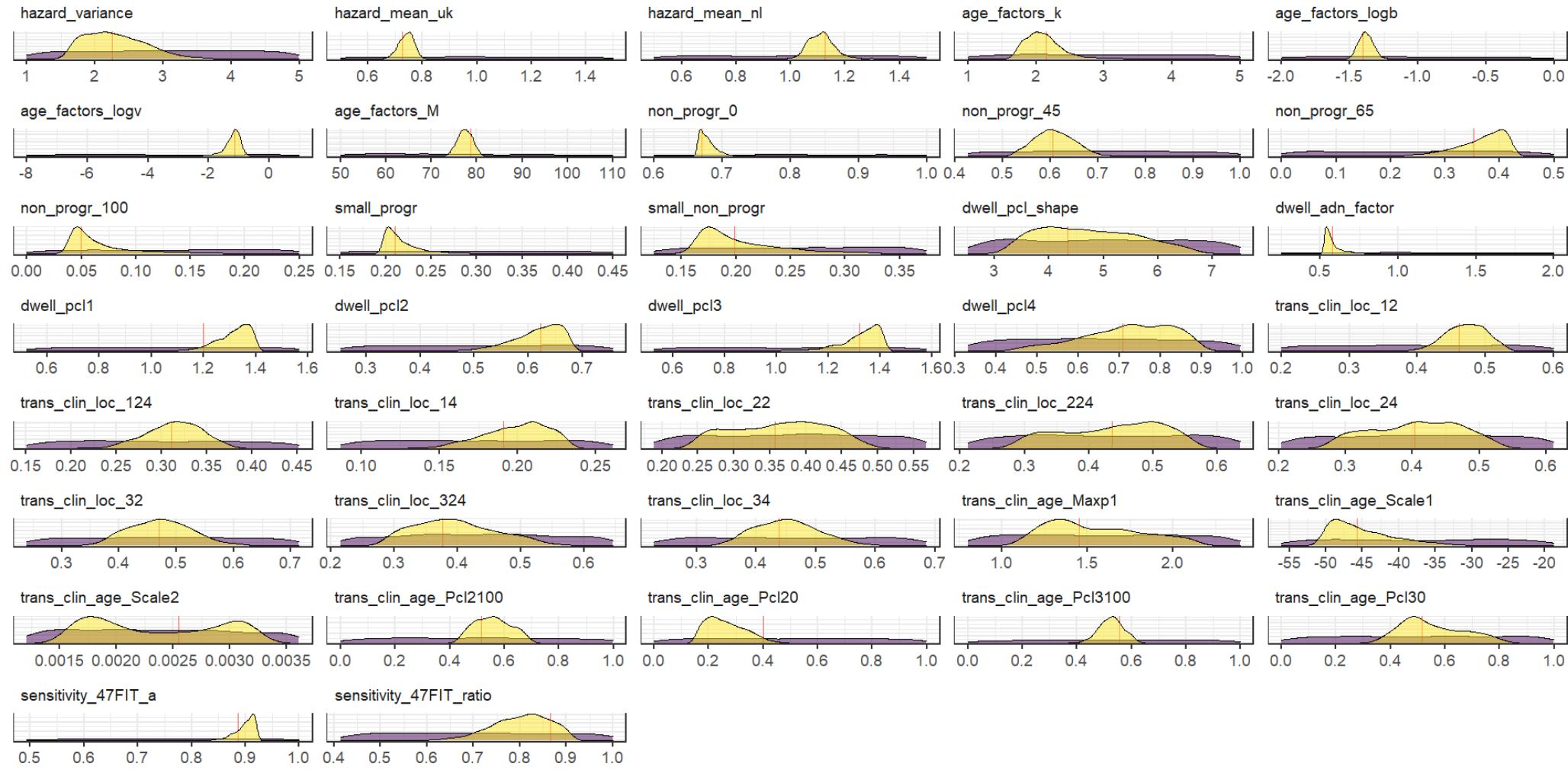


# Results: SimCRC Posterior distributions



Prior      Posterior BayCANN\_SIMCRC

# Results: MISCAN Posterior distributions

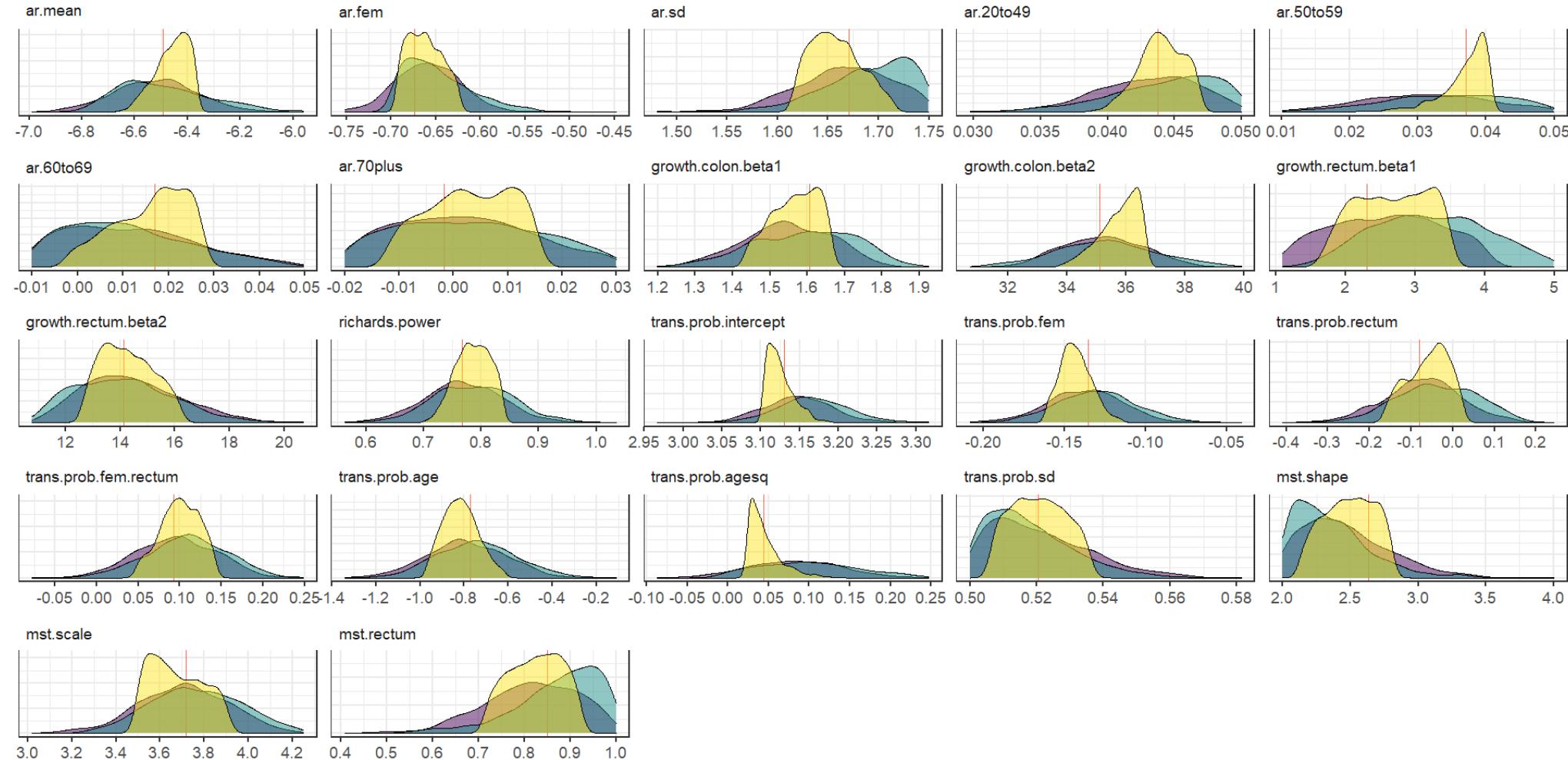


Stan

Prior Posterior BayCANN\_MISCAN

Policy

# Results: CRC SPIN Posterior distributions

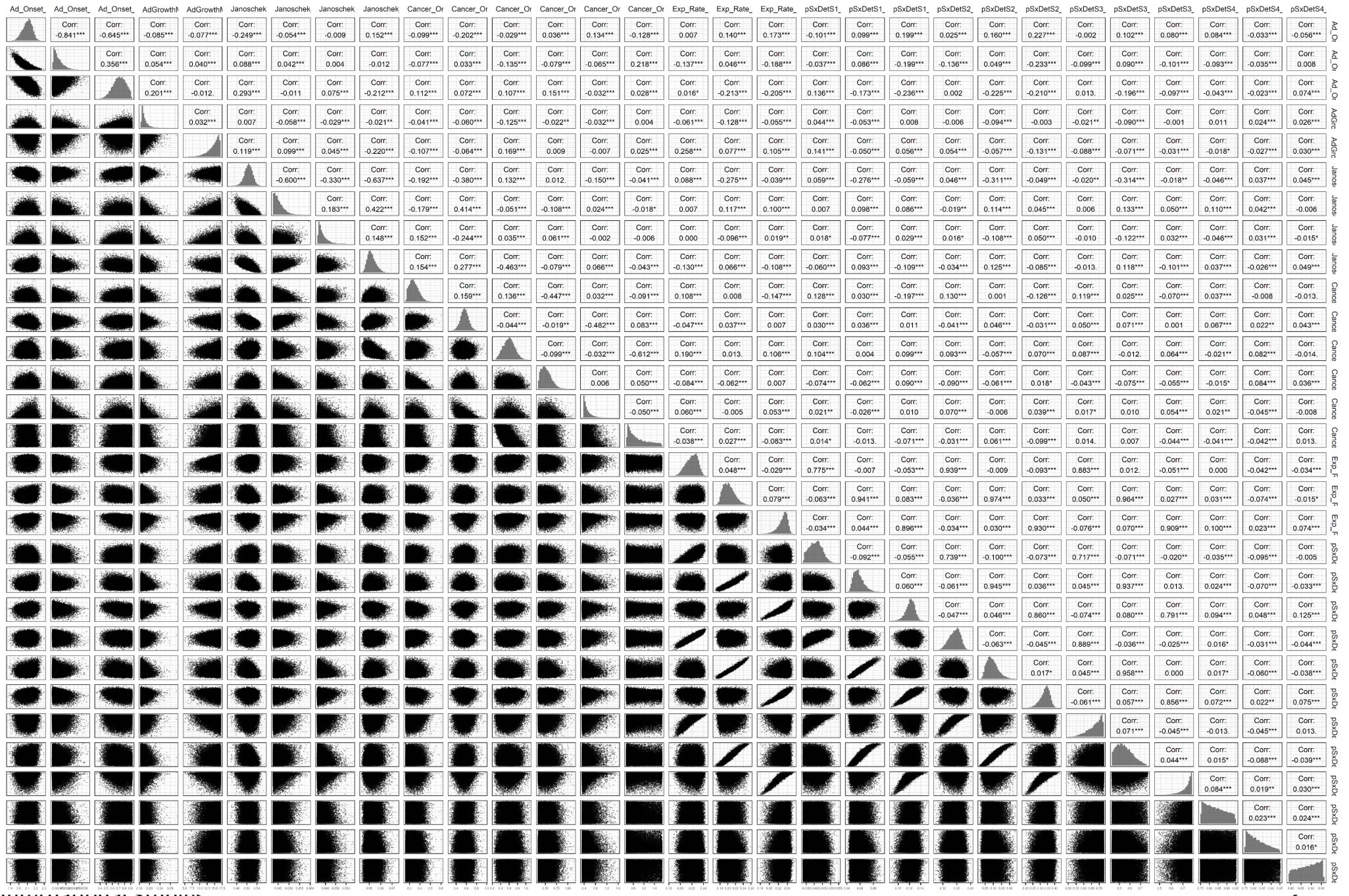


## Pairwise correlations

## Posterior Distributions

SImCRC

Stanford | Fre  
for international studies

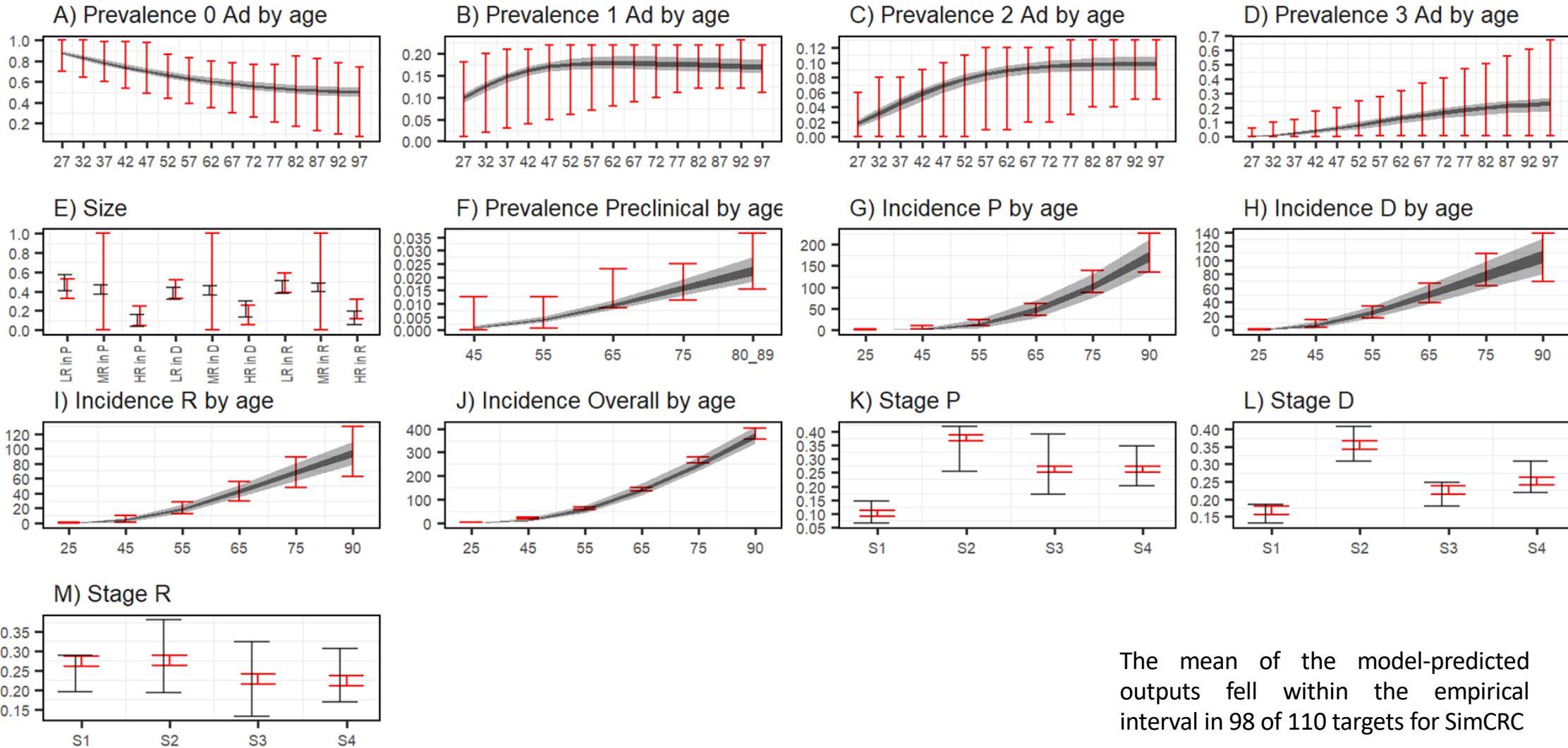


# Results: Calibration time

Model	Required transformation	Structure	Training time	Calibration time	Total time
<b>SimCRC</b>					
<i>30 parameters 110 targets</i>	- Normalize inputs $\sim N(0,1)$ - Scale outputs (-1,1)	- 3 hidden layers - 450 hidden nodes - Activation function: tanh	39 min	105 min	144 min
<b>MISCAN</b>					
<i>37 parameters 84 targets</i>	- Normalize inputs $\sim N(0,1)$ - Outputs standardized by targets	- 1 hidden layer - 600 hidden nodes - Activation function: tanh	23 min	62 min	85 min
<b>CRC SPIN</b>					
<i>22 parameters 41 targets</i>	- Normalize inputs $\sim N(0,1)$ - Scale outputs (-1,1)	- 1 hidden layer - 140 hidden nodes - Activation function: sigmoid	10 min	14 min	24 min

# Results: SimCRC Internal validation

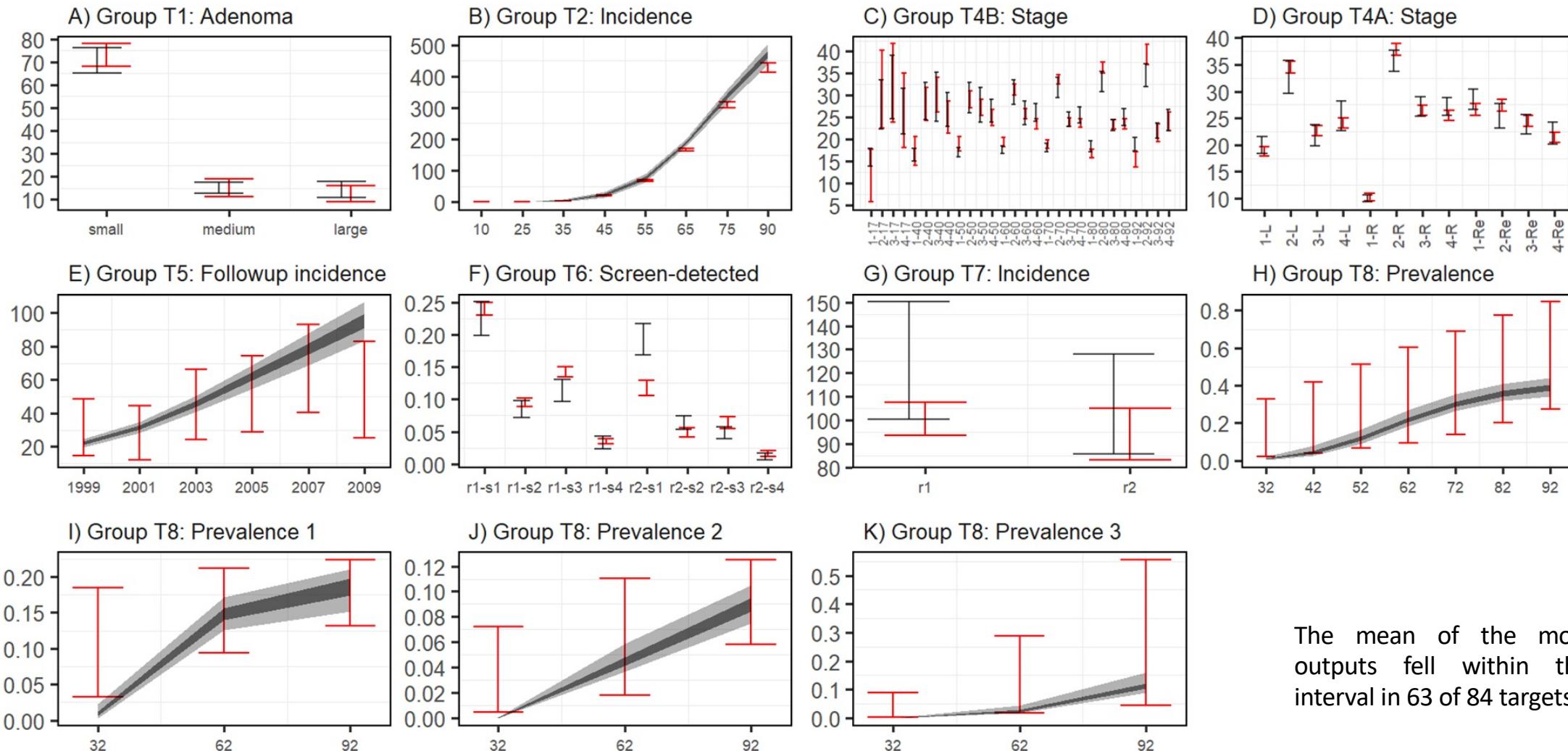
## BayCANN - SimCRC



The mean of the model-predicted outputs fell within the empirical interval in 98 of 110 targets for SimCRC

# Results: MISCAN Internal validation

## BayCANN - MISCAN



The mean of the model-predicted outputs fell within the empirical interval in 63 of 84 targets for MISCAN

In summary

Metamodeling





Thank you!

Fernando Alarid-Escudero [falarid@stanford.edu](mailto:falarid@stanford.edu)

Jeremy Goldhaber-Fiebert [jeremygf@stanford.edu](mailto:jeremygf@stanford.edu)

Jorge Roa [jorgeroa@stanford.edu](mailto:jorgeroa@stanford.edu)