www.ispor.org

ISPOR

Improving healthcare decisions

Model Calibration in R

Presented by:

Eva A. Enns, PhD, MS
Fernando Alarid-Escudero, PhD

ISPOR Short Course Program 2023 (Virtual)

Tuesday and Wednesday, 25-26 July 2023

Learning Objectives

At the end of this workshop, participants will gain an understanding of the following:

- Concept of model calibration and when it is needed
- Steps and decisions involved in setting up a model calibration
- Different model calibration algorithms and their strengths and weaknesses
- Bayesian model calibration (day 2)
- Implementation of a model calibration in R
  - Can be adapted to calibrate your own models (if using R)
  - Can use as a template for any programming language

The DARTH Workgroup

- Materials for this workshop were developed in part by the Decision Analysis in R for Technologies in Health (DARTH) Workgroup
- For more information
  - www.darthworkgroup.com
  - Twitter: @DARTHworkgroup

Why R?

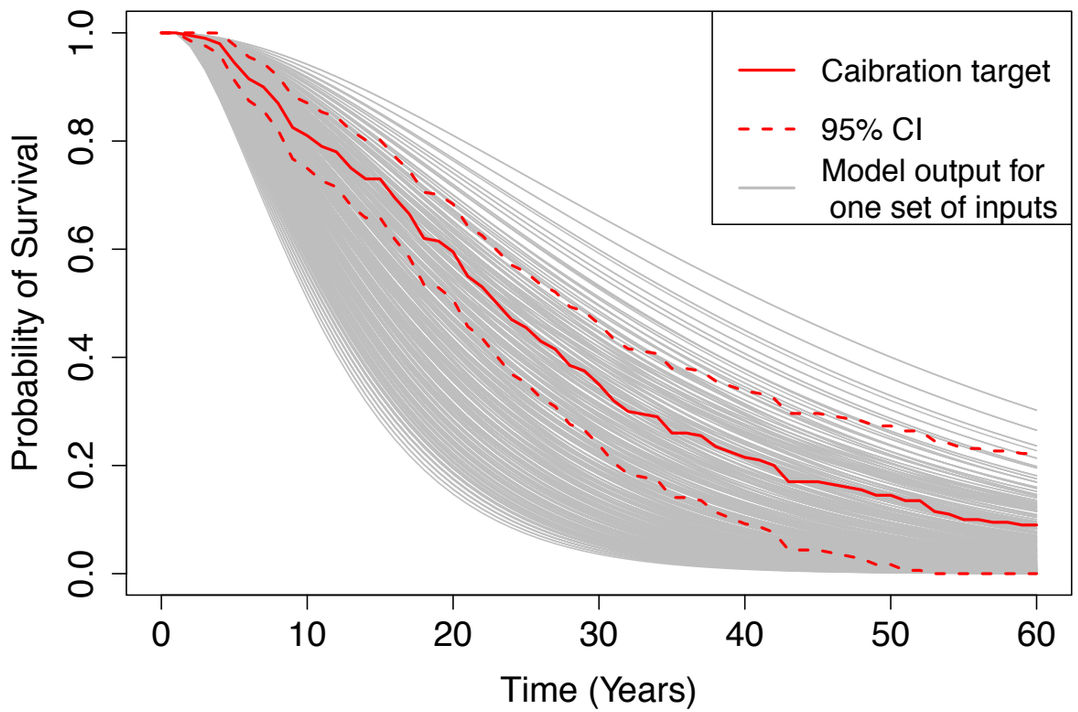
- Open-source, freely available
- Flexible
- Computationally efficient
- Existing packages that implement needed algorithms
- Many other programming languages have these advantages!
  - Can use the calibration framework presented here to write code in your favorite language

Motivation

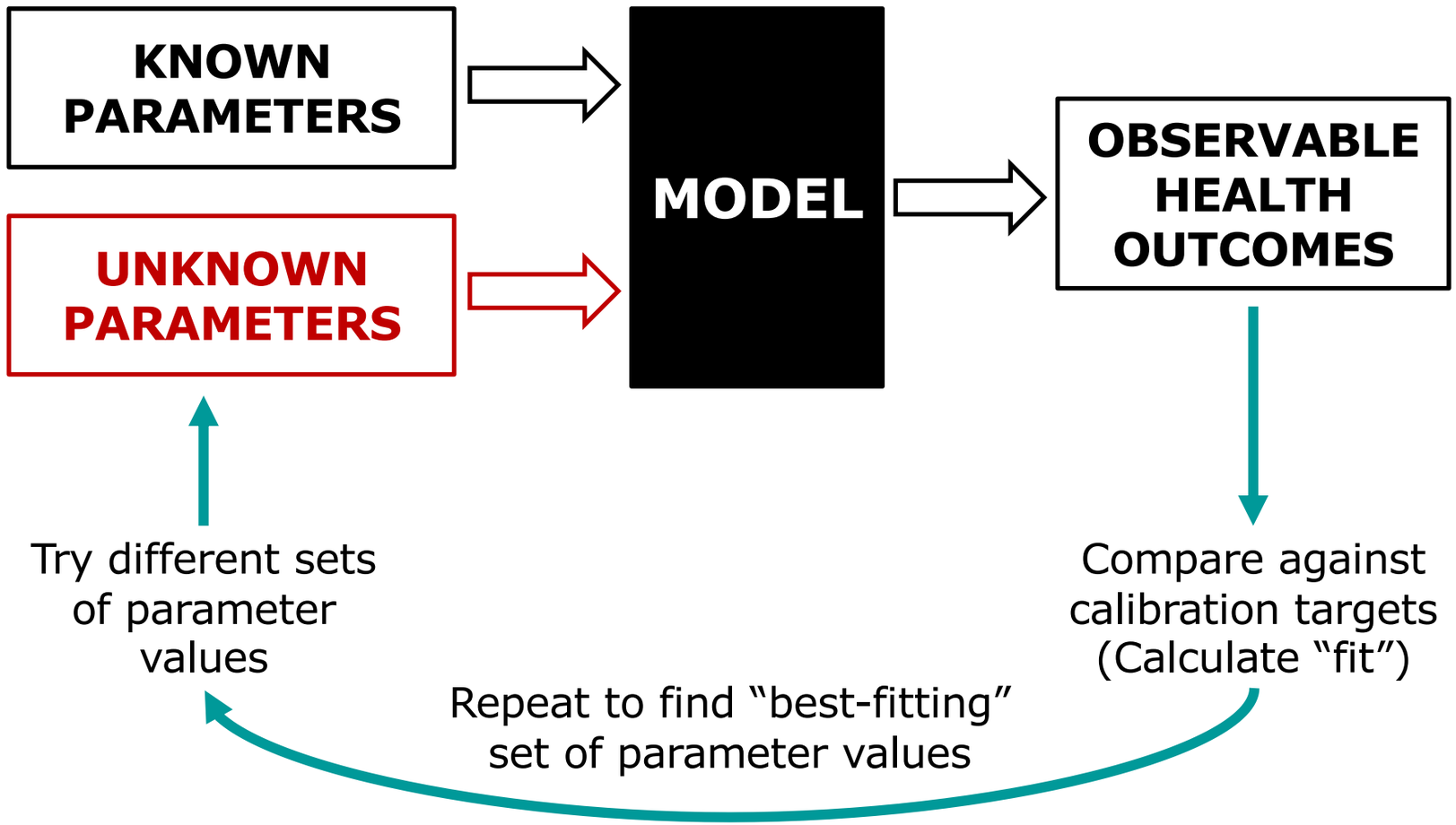
- Mathematical models of disease often involve a subset of parameters whose values are unknown
- Common reasons include physical, feasibility, and ethical limitations
- Estimate values for these parameters by matching model outputs to observed outcomes
  - Model calibration

Calibration definition

- Process of adjusting model input parameter values to match data on an outcome of interest (e.g., survival, prevalence, or incidence)
- Outcome(s) of interest = “calibration targets”



Calibration process



Calibration targets

- Empirical data to be replicated by the model
- Summary statistics (e.g. mean age of cancer diagnosis) or series of observations (e.g. age-specific incidence)
- Can calibrate to multiple targets (e.g. survival and prevalence) simultaneously
- Model should be able to output the outcomes of interest

Calculating “fit”

- Goodness-of-Fit (GoF) is the quantitative measure of how the model is replicating the target data
- Different ways to measure GoF
  - Distance
  - Likelihood
- Notation
  - M : a mathematical model (e.g., Markov model)
  - θ : Set of K parameters to be calibrated
  - $\phi = M(\theta)$: Model output for parameter set θ
  - y : Values of T calibration targets

Distance GoF measures

- Sum of squared errors

$$SSE(\theta) = \sum_{i=1}^{T} \left(y_i - M_i(\theta)\right)^2$$

- Weighted sum of squared errors
  - Assign different weight to different targets (w_i)
  - Often, $w_i = \frac{1}{\sigma^2}$

$$WSSE(\theta) = \sum_{i=1}^{T} w_i \left(y_i - M_i(\theta)\right)^2$$

Likelihood as GoF

- How likely is the observed data to have come from a model M with set of parameter values θ ?
- Assuming targets are independent, overall likelihood is the product of individual likelihoods

$$L(y|M(\theta)) = \prod_{i=1}^{T} L_i(y_i|M(\theta))$$

- Generally, we work with log-likelihood

$$\mathcal{L}(y|M(\theta)) = \sum_{i=1}^{T} \log L_i(y_i|M_i(\theta)$$

Commonly used likelihoods

- **Normal distribution**

$$\log L(y_i, \sigma_i | M_i(\theta)) = -\frac{1}{2}\log\left(2\pi\sigma_i^2\right) - \frac{1}{\sigma_i^2}\left(y_i - M_i(\theta)\right)^2$$

  - In R: `dnorm(x=`y_i`, mean=`$M_i(\theta)$`, sd=`σ_i`, log=T)`

- **Binomial distribution**

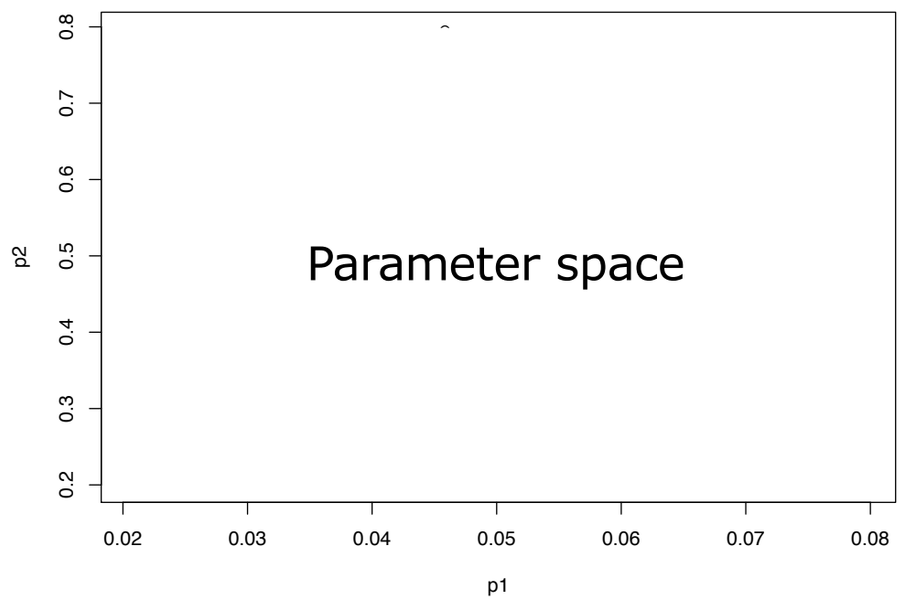
$$\log L(y_i, n_i | M_i(\theta)) = \log\left(\binom{n_i}{y_i}\right) + y_i \log\left(M_i(\theta)\right) + (n_i - y_i)\log\left(1 - M_i(\theta)\right)$$

  - In R: `dbinom(x=`y_i`, size=`n_i`, prob=`$M_i(\theta)$`, log=T)`

- **Multinomial distribution**
  - In R: `dmultinom(x=`y_i`, prob=`$M_i(\theta)$`, log=T)`

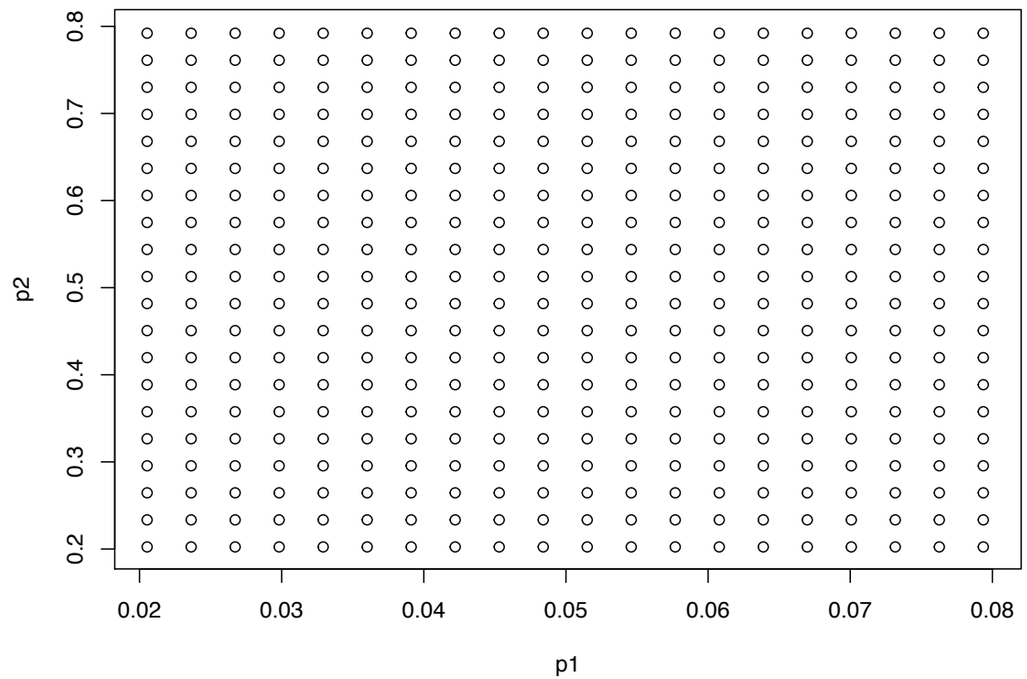
Search Strategy

- Define plausible ranges for parameter whose values are unknown
- Use a search strategy to “search” through the input parameter space
  - Run the model for sets of parameter values generated by search strategy



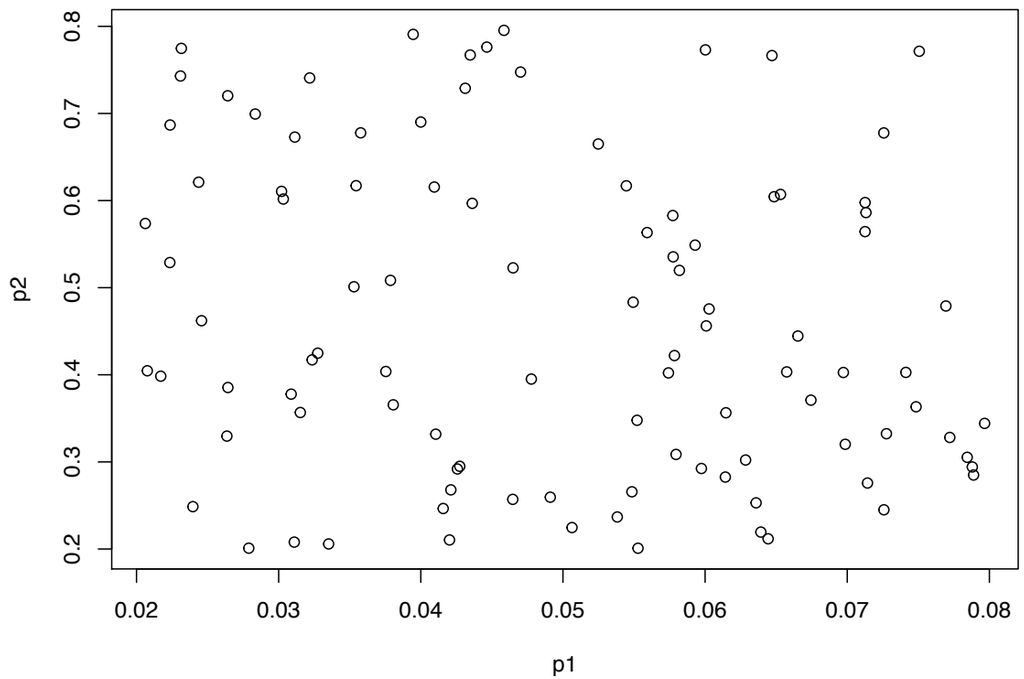
Grid Search

- Run model for all possible combinations of parameter values
- Often infeasible



Random Search

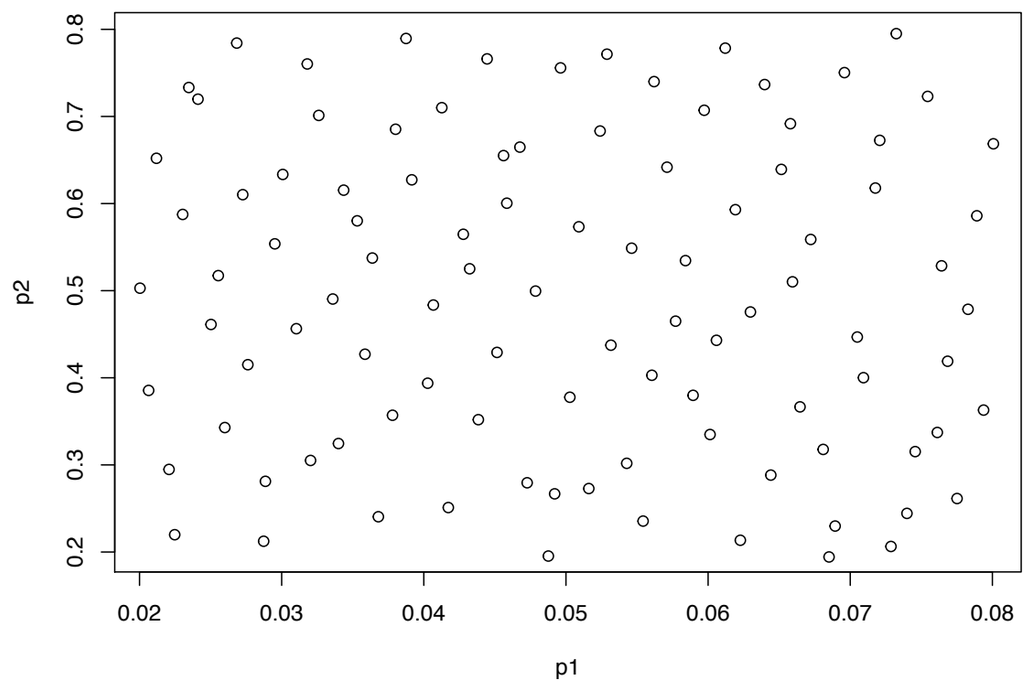
- Randomly sample a large number of parameter value sets from probabilistic distributions
- Use "Latin hypercube sampling" (LHS) to ensure sample captures the full parameter space



Random sample

Random Search

- Randomly sample a large number of parameter value sets from probabilistic distributions
- Use "Latin hypercube sampling" (LHS) to ensure sample captures the full parameter space



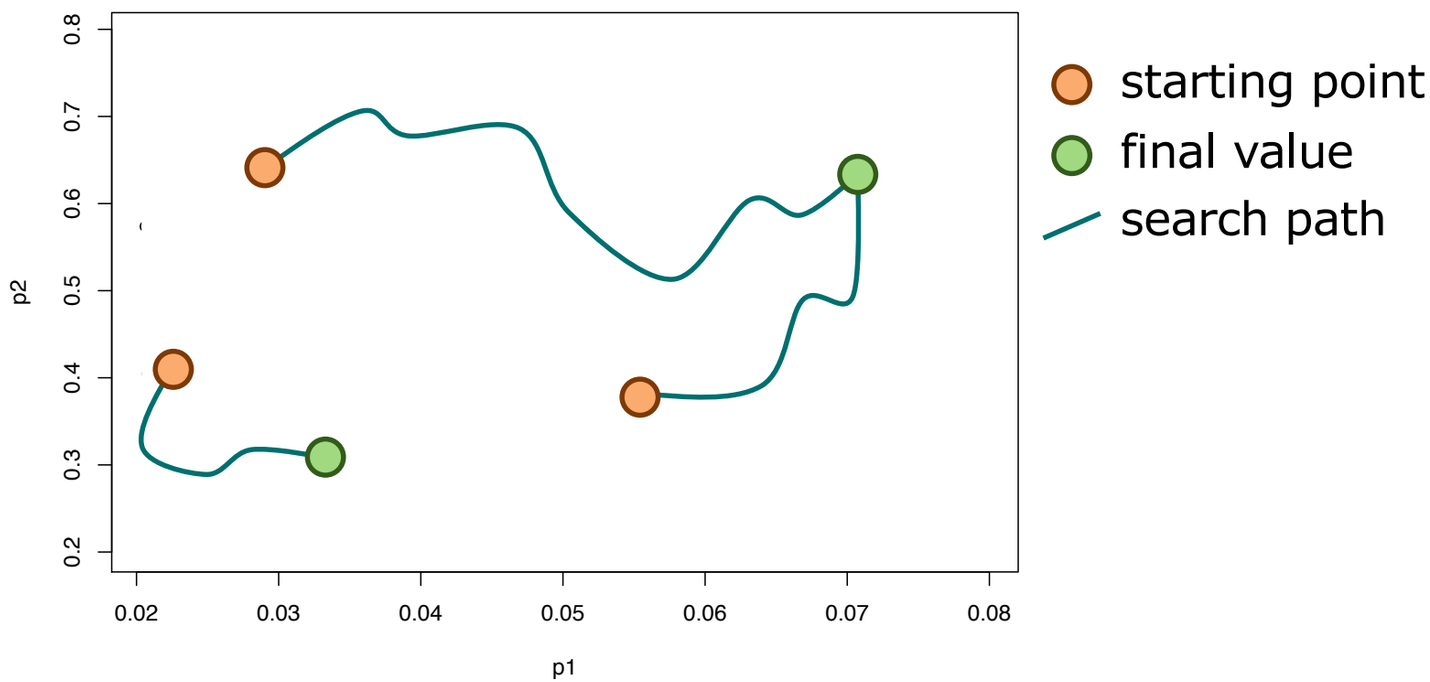
LHS sample

Iterative Search

- Use fits of past input values to determine which input values to try next
- Directed methods
  - Nelder-Mead (simplex method)
  - Gradient–descent and others
- Meta-heuristic algorithms
  - Genetic algorithms
  - Simulated annealing

Nelder-Mead Algorithm

- Downhill simplex method
- Must be run multiple times for different starting points to avoid local extrema



Example:
Calibrating a 3-state cancer model

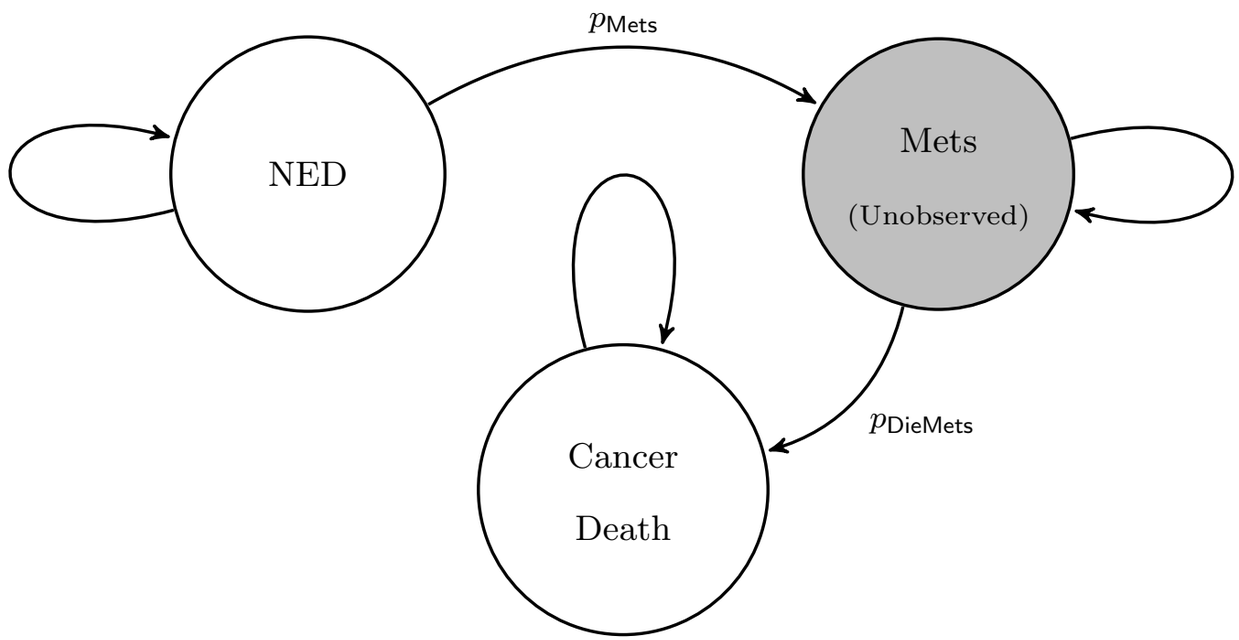
3-state cancer model

- Relative survival as reported by the Surveillance, Epidemiology, and End Results (SEER) Program, represents cancer survival in the absence of other causes of death
- Cancer Markov models often have a distant metastasis state, a state not directly observed in SEER, from which cancer deaths are presumed to occur
- These models are used to model interventions that affect transitions to and from this state

Alarid-Escudero F, Maclehose RF, Peralta Y, Kuntz KM, & Enns, EA. Non-identifiability in model calibration and implications for medical decision making. *Medical Decision Making,* 2018;38(7):810–21.

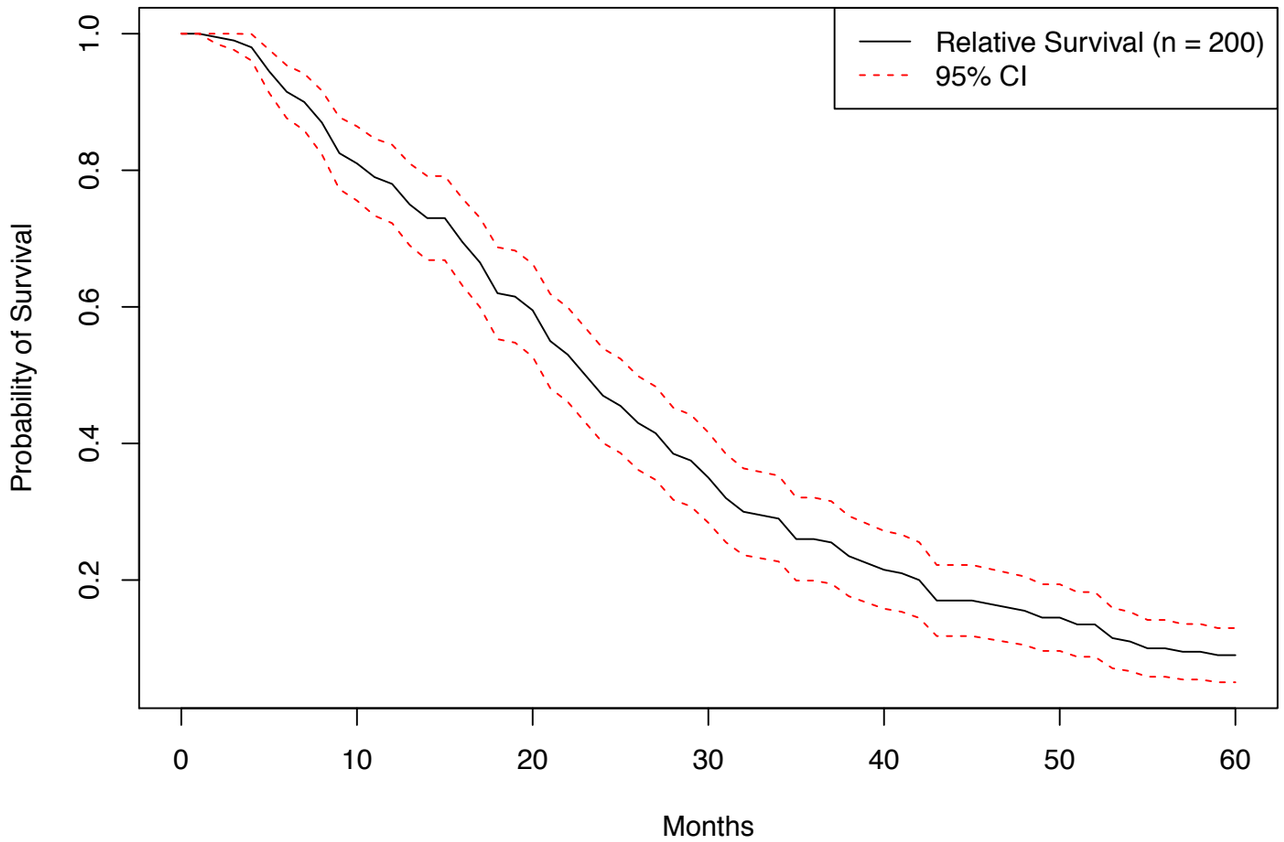
3-state cancer model

- Two unknown transition probabilities
  - p.Mets: Monthly risk of developing distant recurrence, range = [0.04, 0.16]
  - p.DieMets: Monthly risk of cancer death, range = [0.04, 0.16]



Target: Relative survival

- Data frame “CRS_targets$Surv” stored in data file “CRS_CalibTargets.RData”



Calibration R code template

```
### Load calibration targets ###


### Load model as a function ###


### Specify calibration parameters ###


### Calibrate ###


### Explore best-fitting input sets ###
```

R Session

Bayesian Calibration

Bayesian setup

- Instead of a single best-fitting value, we would like an estimate of uncertainty in model parameters, θ, given our observed targets, y
  - E.g. a posterior distribution $p(\theta|y)$
- Recall Bayes theorem

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)}$$

  - $p(\theta)$ is the prior distribution
  - $p(y|\theta)$ has a "related" likelihood function $L(y|\theta) \propto p(y|\theta)$
  - $p(y) = \int p(y|\theta)p(\theta)\, d\theta$ is not a function of θ and often difficult to calculate

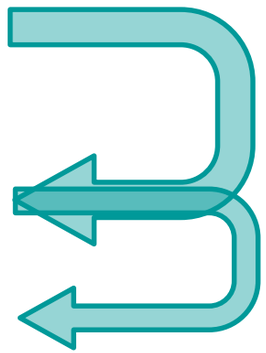
Bayesian setup

- A few steps of math...

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{\cancel{p(y)}}$$

$$p(\theta|y) \propto p(y|\theta)p(\theta)$$

$$p(\theta|y) \propto L(y|\theta)p(\theta)$$



- Using this fact, we can sample the posterior distribution using the likelihood function, prior distribution, and some clever algorithms

Commonly used prior distributions for sampling n_s values

- **Normal distribution**

  In R: `rnorm(n=` n_s`, mean=`$\bar{\theta}$`, sd=`σ_i`)`

- **Uniform distribution**

  In R: `runif(n=` n_s`, min=` lb`, max=`ub`)`

- **Beta distribution**

  In R: `rbeta(n=` n_s`, shape1=`α`, shape2=`β`)`

- **Gamma distribution**

  In R: `rgamma(n=` n_s`, shape=`α`, rate=`β`)`

Pros and Cons of Bayesian Calibration

- **Pros:** We obtain distributions of parameters rather than only point estimates
- **Cons:** “Harder” to implement than other methods? Mmm not really!

Obtaining a Posterior Distribution

- Analytically
  - Not feasible for simulation models
- Markov chain Monte Carlo (MCMC)
- Sampling methods
  - Sampling Importance Resampling (SIR)
  - Incremental Mixture Importance Sampling (IMIS)

Markov chain Monte Carlo (MCMC)

- Simulates posterior distribution by **jumping** in the parameter space in a Markovian fashion
- The transition matrix of the Markov chain is proposed using a **proposal distribution**
- There are numerous algorithms that implement MCMC, including **Metropolis-Hastings** (MH)
- R has some packages that implement MCMC algorithms
- Other R packages interface with external MCMC software, including WinBUGS and JAGS

Markov chain Monte Carlo (MCMC)

- **Pros:**
  - Theoretically will eventually converge to the “true” posterior distribution

- **Cons:**
  - Computationally intensive
  - Autocorrelation is a problem with simulation models and therefore requires a high number of samples
  - Use of external software (WinBUGS, JAGS), requires implementing the model within that software’s syntax

Sampling Importance Resampling (SIR)

Used to simulate posterior distributions (Rubin 1988) with three basic steps

1. **Sampling:** Sample a large number, N, of parameter sets from prior distributions
2. **Importance:**
   1. For each parameter set θ_i $(i = 1, \ldots, N)$, run the simulation model and compute the likelihood
   2. Compute the (normalized) sampling importance weights w_i for each parameter set θ_i by dividing its likelihood value by the sum of likelihood values of all parameter sets,

$$w_i = \frac{L_i}{\sum_{i=1}^{N} L_i}$$

3. **Resampling:** Sample from the discrete distribution of $\{\theta(1), \ldots \theta(N)\}$ with probabilities w_i

Incremental Mixture Importance Sampling (IMIS)

- SIR can miss areas of high posterior probability as the number of parameter increases
- IMIS addresses limitations of SIR and was proposed by Steele at al. (2006)
- Starts with a modest-size SIR
- But in addition to SIR samples, add in samples from a multivariate normal distribution centered at the point with the highest importance weight
- Recalculate importance weights and sample a new sample of input parameter sets
- At the end, posterior becomes a mixture of multivariate normal distributions and of the prior distribution

Incremental Mixture Importance Sampling (IMIS)

1. **Initialization.**
   a) Sample N parameter sets from prior distribution
   b) For each parameter set, θ_i, calculate the likelihood L_i and compute the importance weights w_i
2. **Importance sampling**. Iterate for $k = 1, 2, \ldots$
   a) Choose parameter set with maximum importance weight as the mean of a multivariate normal (MVN) distribution, and compute covariance matrix from nearby parameter sets. Define $H_k = \mathrm{MVN}(\theta^k, \Sigma^k)$.
   b) Sample B new input sets from H_k
   c) Re-calculate importance weights, w_i^k, for all $N + Bk$ samples
   d) Repeat Step 2 until stopping criteria is met.
3. **Resampling**. Resample B_{re} input sets with replacement using importance weights from the last iteration, K. This is a sample from the posterior distribution.

Incremental Mixture Importance Sampling (IMIS) in `R`

- Function `IMIS()` from `IMIS` library takes the following inputs
  - `B`: sample size at each iteration of IMIS
  - `B.re`: number of draws from the posterior
  - `number_k`: maximum number of iterations in IMIS

- IMIS() also requires the following functions to be defined by the user
  - `prior(x)`: returns the prior density of x
  - `likelihood(x)`: returns the likelihood of x
  - `sample.prior(n)`: returns n samples from prior distribution of x

Incremental Mixture Importance Sampling (IMIS) Outputs

- `resamples`: `B.re` draws from the posterior distribution
- `stat`: diagnostic statistics at each IMIS iteration
  - `MargLike`:
  - `UniquePoint`: expected number of unique points among `resamples`
  - `MaxWeight`: maximum importance weight
  - `ESS`: effective sample size (the closer to `B.re`, the better)
- `center`: center of Gaussian components

R Session

DARTH Workgroup

Fernando Alarid-Escudero, PhD[1]
Eva A. Enns, MS, PhD[2]
M.G. Myriam Hunink, MD, PhD[3,4]
Hawre J. Jalal, MD, PhD[5]
Eline M. Krijkamp, MSc[3]
Petros Pechlivanoglou, PhD[6]
Alan Yang, BSc[6]

In collaboration of:
1 Division of Public Administration, Center for Research and Teaching in Economics (CIDE), Aguascalientes, Mexico
2 University of Minnesota School of Public Health, Minneapolis, MN, USA
3 Erasmus MC, Rotterdam, The Netherlands
4 Harvard T.H. Chan School of Public Health, Boston, USA
5 University of Pittsburgh Graduate School of Public Health, Pittsburgh, PA, USA
6 The Hospital for Sick Children, Toronto and University of Toronto, Toronto ON, Canada

www.darthworkgroup.com

CIDE

SCHOOL OF PUBLIC HEALTH
UNIVERSITY OF MINNESOTA

Erasmus MC
Netherlands Institute for Health Sciences

University of Pittsburgh
Graduate School of Public Health

SickKids

© 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.