Employing various calibration methods

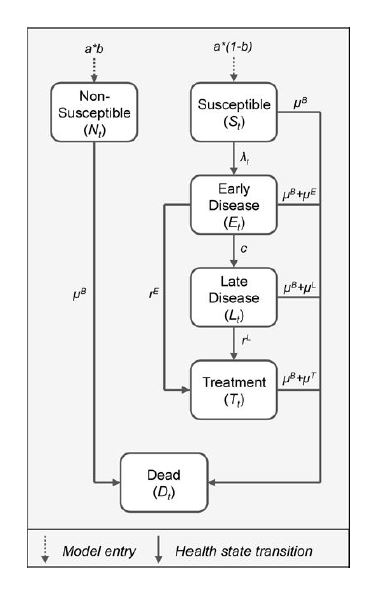
Wael Mohammed

21/04/2022

# Introduction:

## Model: Infectious disease model

The model (Nicolas *et al.,*, 2017) is adapted from approaches for modelling HIV in high-burden settings. The population is divided into five health states including non-susceptible (N), susceptible (S), early disease (E), late disease (L), and treatment (T). The number of individuals by state and year (t) is given by , , , , and respectively. Individuals enter the model distributed across the and states, and transition between states to allow for infection ( to ), disease progression ( to ), treatment initiation ( and to ), and death (, , , and to ) via background and disease-specific mortality. The diagram below represents the model.



HIV model

## Calibration process specifications:

### Calibration parameters:

* mu\_e Cause-specific mortality rate with early-stage disease
* mu\_l Cause-specific mortality rate with late-stage disease
* mu\_t Cause-specific mortality rate on treatment
* p Transition rate from early to late-stage disease
* r\_l Rate of uptake onto treatment (r\_l = late-stage disease)
* rho Effective contact rate
* b Fraction of population in at-risk group

### Calibration targets:

* Prev Prevalence at 10, 20, 30 years
* Surv HIV survival without treatment
* Trt\_vol Treatment volume at 30 years

### Parameter transformations:

Two calibration process was carried out twice, with and without parameter-transformation.

The calibration parameters were transformed as follows: - b was transformed to the entire real line using logit. - All other parameters were transformed using log.

### Search method:

* Random search using:
  + Full factorial grid
  + Random grid
  + Latin-Hypercube Sampling
* Directed search using:
  + Gradient-based (GB) (uses derivatives)
  + Nelder-Mead (NM) algorithm, aka simplex method (derivative free)
  + Global optimization techniques:
    - Simulated Annealing (SANN)
    - Genetic algorithms (GA)

### Goodness-of-fit measure:

* Sum of log-likelihoods
* Sum of squared errors

### Bayesian methods:

* Sampling Importance Resampling (SIR)
* Incremental Mixture Importance Sampling (IMIS)

# The Calibration process:

The code blocks below show the code for the untransformed version of the model; while the results from both runs will be displayed below in the results sub-section.

## Sampling from prior using different methods:

set.seed(1)  
HID\_results <- list()  
# LHS:----  
HID\_results$Prior\_samples[['LHS']] <- sample\_prior\_LHS(  
 .n\_samples = 10000,  
 .l\_params = HID\_data$l\_params)  
# FGS:----  
HID\_results$Prior\_samples[['FGS']] <- sample\_prior\_FGS(  
 .n\_samples = 10000,  
 .l\_params = HID\_data$l\_params)  
# RGS:----  
HID\_results$Prior\_samples[['RGS']] <- sample\_prior\_RGS(  
 .n\_samples = 10000,  
 .l\_params = HID\_data$l\_params)

### A quick look at a sample of the samples:

Values from both runs of the process are shown below.

# Original scale parameters:----  
HID\_results$Prior\_samples[['LHS']] %>%  
 head(5)

## # A tibble: 5 x 7  
## mu\_e mu\_l mu\_t p r\_l rho b  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0.0644 0.289 0.0136 0.0837 0.528 0.265 0.320   
## 2 0.0744 0.265 0.0162 0.118 0.417 0.613 0.0581  
## 3 0.0437 0.257 0.0271 0.214 0.281 0.312 0.0717  
## 4 0.0334 0.259 0.0278 0.0723 0.410 0.579 0.161   
## 5 0.0288 0.372 0.0102 0.102 0.503 0.493 0.238

# Transformed scale parameters:----  
HID2\_results$Prior\_samples[['LHS']] %>%  
 head(5)

## # A tibble: 5 x 7  
## mu\_e mu\_l mu\_t p r\_l rho b  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 -2.74 -1.24 -4.30 -2.48 -0.638 -1.33 0.571  
## 2 -2.60 -1.33 -4.12 -2.14 -0.876 -0.490 -4.02   
## 3 -3.13 -1.36 -3.61 -1.54 -1.27 -1.17 -3.60   
## 4 -3.40 -1.35 -3.58 -2.63 -0.892 -0.547 -1.70   
## 5 -3.55 -0.990 -4.58 -2.29 -0.688 -0.708 -0.511

## Using random-search with goodness-of-fit (without optimisation):

While any of the three random search methods can be used, the following code uses LHS.

# LHS with weighted sum of square errors:----  
HID\_results$Calib\_results$Random[[1]] <- wSSE\_GOF(  
 .func = HID\_markov, .optim = FALSE,  
 .args = NULL,  
 .samples = HID\_results$Prior\_samples$LHS,  
 .l\_targets = HID\_data$l\_targets,  
 .sample\_method = "LHS")  
# LHS with log likelihood:----  
HID\_results$Calib\_results$Random[[2]] <- LLK\_GOF(  
 .func = HID\_markov, .optim = FALSE,  
 .args = NULL,  
 .samples = HID\_results$Prior\_samples$LHS,  
 .l\_targets = HID\_data$l\_targets,  
 .sample\_method = "LHS")

### A quick look at a sample of the calibration results:

Values from runs using original and scaled parameters are shown below.

# O\_scale: LHS with weighted sum of square errors:----  
HID\_results$Calib\_results$Random[[1]] %>%  
 head(5)

## # A tibble: 5 x 9  
## mu\_e mu\_l mu\_t p r\_l rho b Overall\_fit Label   
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <chr>   
## 1 0.0289 0.145 0.0238 0.125 0.618 0.531 0.194 -2.61 wSumSquareError\_LHS  
## 2 0.0699 0.104 0.0307 0.198 0.532 0.565 0.225 -5.83 wSumSquareError\_LHS  
## 3 0.0229 0.178 0.0195 0.130 0.689 0.564 0.178 -6.84 wSumSquareError\_LHS  
## 4 0.0219 0.235 0.0190 0.147 0.558 0.509 0.189 -8.93 wSumSquareError\_LHS  
## 5 0.0192 0.297 0.0271 0.130 0.901 0.508 0.192 -10.1 wSumSquareError\_LHS

# O\_scale: LHS with log likelihood:----  
HID\_results$Calib\_results$Random[[2]] %>%  
 head(5)

## # A tibble: 5 x 9  
## mu\_e mu\_l mu\_t p r\_l rho b Overall\_fit Label   
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <chr>   
## 1 0.0289 0.145 0.0238 0.125 0.618 0.531 0.194 -18.9 log\_likelihood\_LHS  
## 2 0.0229 0.178 0.0195 0.130 0.689 0.564 0.178 -19.7 log\_likelihood\_LHS  
## 3 0.0699 0.104 0.0307 0.198 0.532 0.565 0.225 -20.4 log\_likelihood\_LHS  
## 4 0.0219 0.235 0.0190 0.147 0.558 0.509 0.189 -21.2 log\_likelihood\_LHS  
## 5 0.0191 0.212 0.0148 0.0885 0.926 0.555 0.178 -21.3 log\_likelihood\_LHS

# T\_scale: LHS with log likelihood:----  
HID2\_results$Calib\_results$Random[[1]] %>%  
 head(5)

## # A tibble: 5 x 9  
## mu\_e mu\_l mu\_t p r\_l rho b Overall\_fit Label   
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <chr>   
## 1 -2.55 -2.72 -4.36 -2.41 -0.330 -0.589 -1.28 -5.06 wSumSquareError\_LHS  
## 2 -3.05 -1.92 -3.60 -2.04 -0.806 -0.592 -1.17 -6.48 wSumSquareError\_LHS  
## 3 -5.15 -1.10 -4.27 -2.07 -0.555 -0.695 -1.39 -7.27 wSumSquareError\_LHS  
## 4 -3.34 -1.21 -4.02 -1.80 -0.796 -0.579 -1.21 -10.0 wSumSquareError\_LHS  
## 5 -3.29 -2.03 -3.79 -1.91 -1.44 -0.721 -1.17 -12.1 wSumSquareError\_LHS

# T\_scale: LHS with log likelihood:----  
HID2\_results$Calib\_results$Random[[2]] %>%  
 head(5)

## # A tibble: 5 x 9  
## mu\_e mu\_l mu\_t p r\_l rho b Overall\_fit Label   
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <chr>   
## 1 -3.05 -1.92 -3.60 -2.04 -0.806 -0.592 -1.17 -19.6 log\_likelihood\_LHS  
## 2 -5.15 -1.10 -4.27 -2.07 -0.555 -0.695 -1.39 -19.9 log\_likelihood\_LHS  
## 3 -2.55 -2.72 -4.36 -2.41 -0.330 -0.589 -1.28 -20.2 log\_likelihood\_LHS  
## 4 -3.29 -2.03 -3.79 -1.91 -1.44 -0.721 -1.17 -20.8 log\_likelihood\_LHS  
## 5 -3.50 -2.01 -3.42 -2.27 -0.732 -0.614 -1.21 -22.1 log\_likelihood\_LHS

## Optimisation algorithms:

Both goodness-of-fit methods are used in the code below.

# Sample 5 starting parameter sets for the optimisation algorithms:----  
HID\_results$Prior\_samples[['LHS\_Directed']] <- sample\_prior\_LHS(  
 .n\_samples = 5,  
 .l\_params = HID\_data$l\_params)  
# Nelder-Mead using log\_likelihood as goodness-of-fit:----  
HID\_results$Calib\_results$Directed[[1]] <- calibrateModel\_directed(  
 .l\_params = HID\_data$l\_params,  
 .func = HID\_markov,  
 .args = NULL,  
 .gof = 'log\_likelihood',  
 .samples = HID\_results$Prior\_samples$LHS\_Directed,  
 .s\_method = 'Nelder-Mead',  
 .maximise = TRUE,  
 .l\_targets = HID\_data$l\_targets,  
 maxit = 1000)  
# Nelder-Mead using sum-squared-errors as goodness-of-fit:----  
HID\_results$Calib\_results$Directed[[2]] <- calibrateModel\_directed(  
 .l\_params = HID\_data$l\_params,  
 .func = HID\_markov,  
 .args = NULL,  
 .gof = 'wSumSquareError',  
 .samples = HID\_results$Prior\_samples$LHS\_Directed,  
 .s\_method = 'Nelder-Mead',  
 .maximise = TRUE,  
 .l\_targets = HID\_data$l\_targets,  
 maxit = 1000)  
# Gradient-based using log\_likelihood as goodness-of-fit:----  
HID\_results$Calib\_results$Directed[[3]] <- calibrateModel\_directed(  
 .l\_params = HID\_data$l\_params,  
 .func = HID\_markov,  
 .args = NULL,  
 .gof = 'log\_likelihood',  
 .samples = HID\_results$Prior\_samples$LHS\_Directed,  
 .s\_method = 'BFGS',  
 .maximise = TRUE,  
 .l\_targets = HID\_data$l\_targets,  
 maxit = 1000)  
# Gradient-based using sum-squared-errors as goodness-of-fit:----  
HID\_results$Calib\_results$Directed[[4]] <- calibrateModel\_directed(  
 .l\_params = HID\_data$l\_params,  
 .func = HID\_markov,  
 .args = NULL,  
 .gof = 'wSumSquareError',  
 .samples = HID\_results$Prior\_samples$LHS\_Directed,  
 .s\_method = 'BFGS',  
 .maximise = TRUE,  
 .l\_targets = HID\_data$l\_targets,  
 maxit = 1000)  
# Simulated annealing using log\_likelihood as goodness-of-fit:----  
HID\_results$Calib\_results$Directed[[5]] <- calibrateModel\_directed(  
 .l\_params = HID\_data$l\_params,  
 .func = HID\_markov,  
 .args = NULL,  
 .gof = 'log\_likelihood',  
 .samples = HID\_results$Prior\_samples$LHS\_Directed,  
 .s\_method = 'SANN',  
 .maximise = TRUE,  
 .l\_targets = HID\_data$l\_targets,  
 fnscale = -1,  
 temp = 10,  
 tmax = 10,  
 maxit = 1000)  
# Simulated annealing using sum-squared-errors as goodness-of-fit:----  
HID\_results$Calib\_results$Directed[[6]] <- calibrateModel\_directed(  
 .l\_params = HID\_data$l\_params,  
 .func = HID\_markov,  
 .args = NULL,  
 .gof = 'wSumSquareError',  
 .samples = HID\_results$Prior\_samples$LHS\_Directed,  
 .s\_method = 'SANN',  
 .maximise = TRUE,  
 .l\_targets = HID\_data$l\_targets,  
 maxit = 1000,  
 temp = 10,  
 tmax = 10)  
# Genetic algorithm using log\_likelihood as goodness-of-fit:----  
HID\_results$Calib\_results$Directed[[7]] <- calibrateModel\_directed(  
 .l\_params = HID\_data$l\_params,  
 .func = HID\_markov,  
 .args = NULL,  
 .gof = 'log\_likelihood',  
 .samples = HID\_results$Prior\_samples$LHS\_Directed,  
 .s\_method = 'GA',  
 .maximise = TRUE,  
 .l\_targets = HID\_data$l\_targets,  
 maxit = 1000,  
 temp = 10,  
 tmax = 10)  
# Genetic algorithm using sum-squared-errors as goodness-of-fit:----  
HID\_results$Calib\_results$Directed[[8]] <- calibrateModel\_directed(  
 .l\_params = HID\_data$l\_params,  
 .func = HID\_markov,  
 .args = NULL,  
 .gof = 'wSumSquareError',  
 .samples = HID\_results$Prior\_samples$LHS\_Directed,  
 .s\_method = 'GA',  
 .maximise = TRUE,  
 .l\_targets = HID\_data$l\_targets,  
 maxit = 1000,  
 temp = 10,  
 tmax = 10)

### A quick look at a sample of the calibration results:

# Original scale:----  
## O: Nelder-Mead using log\_likelihood as goodness-of-fit:----  
HID\_results$Calib\_results$Directed[[1]][[1]]

## $Params  
## [1] "mu\_e" "mu\_l" "mu\_t" "p" "r\_l" "rho" "b"   
##   
## $Estimate  
## mu\_e mu\_l mu\_t p r\_l rho b   
## 0.28805344 0.05622027 0.05581779 0.50310717 0.49872033 0.78114047 0.33214270   
##   
## $Lower  
## mu\_e mu\_l mu\_t p r\_l rho b   
## -2.5595985 NaN NaN NaN -1.6747249 -2.0514301 -0.5286404   
##   
## $Upper  
## mu\_e mu\_l mu\_t p r\_l rho b   
## 3.135705 NaN NaN NaN 2.672166 3.613711 1.192926   
##   
## $`GOF value`  
## [1] -18.02995  
##   
## $`Calibration method`  
## [1] "Nelder-Mead\_log\_likelihood"  
##   
## $Sigma  
## mu\_e mu\_l mu\_t p r\_l rho  
## mu\_e 1.00000000 -0.101591904 0.035020949 2.7271847 1.6766966 1.99600575  
## mu\_l -0.10159190 1.000000000 -0.008389984 -0.4052099 0.1734383 -0.06427064  
## mu\_t 0.03502095 -0.008389984 1.000000000 -0.1194201 0.2879930 0.07634126  
## p 2.72718466 -0.405209873 -0.119420064 1.0000000 8.4151945 3.36012082  
## r\_l 1.67669664 0.173438285 0.287993010 8.4151945 1.0000000 2.14818545  
## rho 1.99600575 -0.064270643 0.076341259 3.3601208 2.1481854 1.00000000  
## b 0.62244706 -0.002307483 0.039846218 1.4836514 0.5245887 0.65884741  
## b  
## mu\_e 0.622447064  
## mu\_l -0.002307483  
## mu\_t 0.039846218  
## p 1.483651352  
## r\_l 0.524588688  
## rho 0.658847414  
## b 1.000000000

## O: Nelder-Mead using sum-squared-errors as goodness-of-fit:----  
HID\_results$Calib\_results$Directed[[2]][[1]]

## $Params  
## [1] "mu\_e" "mu\_l" "mu\_t" "p" "r\_l" "rho" "b"   
##   
## $Estimate  
## mu\_e mu\_l mu\_t p r\_l rho b   
## 0.05076314 0.09749395 0.05420446 0.30822384 0.35694242 0.57205487 0.24699899   
##   
## $Lower  
## mu\_e mu\_l mu\_t p r\_l rho   
## -0.719304172 -0.210872101 -0.027553261 -0.124163792 -0.234342225 0.114684680   
## b   
## 0.006428745   
##   
## $Upper  
## mu\_e mu\_l mu\_t p r\_l rho b   
## 0.8208305 0.4058600 0.1359622 0.7406115 0.9482271 1.0294251 0.4875692   
##   
## $`GOF value`  
## [1] -0.08999125  
##   
## $`Calibration method`  
## [1] "Nelder-Mead\_wSumSquareError"  
##   
## $Sigma  
## mu\_e mu\_l mu\_t p r\_l  
## mu\_e 1.00000000 -0.061561327 -0.013863865 -0.040093142 -0.034355899  
## mu\_l -0.06156133 1.000000000 0.005435879 0.014252485 0.014778103  
## mu\_t -0.01386386 0.005435879 1.000000000 0.005308286 0.006909972  
## p -0.04009314 0.014252485 0.005308286 1.000000000 -0.014162234  
## r\_l -0.03435590 0.014778103 0.006909972 -0.014162234 1.000000000  
## rho 0.09026999 -0.036155744 -0.007330294 -0.019507884 -0.013472026  
## b 0.04663242 -0.018736719 -0.003546318 -0.008488797 -0.007677976  
## rho b  
## mu\_e 0.090269987 0.046632421  
## mu\_l -0.036155744 -0.018736719  
## mu\_t -0.007330294 -0.003546318  
## p -0.019507884 -0.008488797  
## r\_l -0.013472026 -0.007677976  
## rho 1.000000000 0.028242510  
## b 0.028242510 1.000000000

## O: Gradient-based using log\_likelihood as goodness-of-fit:----  
HID\_results$Calib\_results$Directed[[3]][[1]]

## $Params  
## [1] "mu\_e" "mu\_l" "mu\_t" "p" "r\_l" "rho" "b"   
##   
## $Estimate  
## mu\_e mu\_l mu\_t p r\_l rho b   
## 0.14612744 0.06978915 0.06201307 0.34001705 0.90991076 0.70838904 0.30975246   
##   
## $Lower  
## mu\_e mu\_l mu\_t p r\_l rho   
## -4.9924981 -0.9322913 -0.2904844 -2.6290137 -37.3678271 -6.4737656   
## b   
## -2.6631467   
##   
## $Upper  
## mu\_e mu\_l mu\_t p r\_l rho b   
## 5.2847529 1.0718696 0.4145106 3.3090478 39.1876486 7.8905437 3.2826516   
##   
## $`GOF value`  
## [1] -18.02993  
##   
## $`Calibration method`  
## [1] "BFGS\_log\_likelihood"  
##   
## $Sigma  
## mu\_e mu\_l mu\_t p r\_l rho  
## mu\_e 1.0000000 -1.33655175 -0.162711208 3.03089504 29.105200 8.78436474  
## mu\_l -1.3365517 1.00000000 0.035135393 -0.55291597 -5.570553 -1.68911869  
## mu\_t -0.1627112 0.03513539 1.000000000 0.02038783 1.749663 0.03396454  
## p 3.0308950 -0.55291597 0.020387828 1.00000000 15.170430 4.37226551  
## r\_l 29.1051999 -5.57055334 1.749663116 15.17043001 1.000000 60.54840567  
## rho 8.7843647 -1.68911869 0.033964541 4.37226551 60.548406 1.00000000  
## b 3.7348677 -0.71822652 -0.001772307 1.85726106 23.845422 5.54153620  
## b  
## mu\_e 3.734867651  
## mu\_l -0.718226516  
## mu\_t -0.001772307  
## p 1.857261055  
## r\_l 23.845421601  
## rho 5.541536200  
## b 1.000000000

## O: Gradient-based using sum-squared-errors as goodness-of-fit:----  
HID\_results$Calib\_results$Directed[[4]][[1]]

## $Params  
## [1] "mu\_e" "mu\_l" "mu\_t" "p" "r\_l" "rho" "b"   
##   
## $Estimate  
## mu\_e mu\_l mu\_t p r\_l rho b   
## 0.05300305 0.09492525 0.06949119 0.35494867 0.70669255 0.62436759 0.27229941   
##   
## $Lower  
## mu\_e mu\_l mu\_t p r\_l rho b   
## -1.7432791 -0.5144934 NaN NaN NaN NaN NaN   
##   
## $Upper  
## mu\_e mu\_l mu\_t p r\_l rho b   
## 1.8492852 0.7043439 NaN NaN NaN NaN NaN   
##   
## $`GOF value`  
## [1] -2.735461e-05  
##   
## $`Calibration method`  
## [1] "BFGS\_wSumSquareError"  
##   
## $Sigma  
## mu\_e mu\_l mu\_t p r\_l rho  
## mu\_e 1.00000000 -0.28679958 -0.15687135 0.03350305 -4.025401 0.2148506  
## mu\_l -0.28679958 1.00000000 0.04545154 0.03285284 0.807722 -0.1166302  
## mu\_t -0.15687135 0.04545154 1.00000000 0.25765555 -2.119228 -0.2480600  
## p 0.03350305 0.03285284 0.25765555 1.00000000 18.287443 1.4178121  
## r\_l -4.02540091 0.80772202 -2.11922802 18.28744329 1.000000 -16.4397707  
## rho 0.21485058 -0.11663016 -0.24805997 1.41781215 -16.439771 1.0000000  
## b 0.16445427 -0.06877564 -0.08711878 0.41806522 -5.071922 -0.2630917  
## b  
## mu\_e 0.16445427  
## mu\_l -0.06877564  
## mu\_t -0.08711878  
## p 0.41806522  
## r\_l -5.07192157  
## rho -0.26309170  
## b 1.00000000

## O: Simulated annealing using log\_likelihood as goodness-of-fit:----  
HID\_results$Calib\_results$Directed[[5]][[1]]

## $Params  
## [1] "mu\_e" "mu\_l" "mu\_t" "p" "r\_l" "rho" "b"   
##   
## $Estimate  
## mu\_e mu\_l mu\_t p r\_l rho b   
## 0.04994421 0.12824889 0.01522709 0.09797706 0.47391692 0.50465331 0.17130402   
##   
## $Lower  
## mu\_e mu\_l mu\_t p r\_l rho b   
## NaN NaN NaN 0.08595155 0.39202023 NaN NaN   
##   
## $Upper  
## mu\_e mu\_l mu\_t p r\_l rho b   
## NaN NaN NaN 0.1100026 0.5558136 NaN NaN   
##   
## $`GOF value`  
## [1] -47.7032  
##   
## $`Calibration method`  
## [1] "SANN\_log\_likelihood"  
##   
## $Sigma  
## mu\_e mu\_l mu\_t p r\_l  
## mu\_e 1.000000e+00 0.0001196046 2.267945e-04 3.015913e-05 0.0003741041  
## mu\_l 1.196046e-04 1.0000000000 5.001789e-04 1.944087e-04 -0.0011760839  
## mu\_t 2.267945e-04 0.0005001789 1.000000e+00 -7.703889e-05 0.0004606983  
## p 3.015913e-05 0.0001944087 -7.703889e-05 1.000000e+00 0.0001516345  
## r\_l 3.741041e-04 -0.0011760839 4.606983e-04 1.516345e-04 1.0000000000  
## rho -2.211008e-04 -0.0003317648 4.509490e-04 8.186478e-05 -0.0001335241  
## b 1.521143e-04 0.0004183804 -2.111413e-04 -5.252147e-05 0.0004003843  
## rho b  
## mu\_e -2.211008e-04 1.521143e-04  
## mu\_l -3.317648e-04 4.183804e-04  
## mu\_t 4.509490e-04 -2.111413e-04  
## p 8.186478e-05 -5.252147e-05  
## r\_l -1.335241e-04 4.003843e-04  
## rho 1.000000e+00 2.481076e-04  
## b 2.481076e-04 1.000000e+00

## O: Simulated annealing using sum-squared-errors as goodness-of-fit:----  
HID\_results$Calib\_results$Directed[[6]][[1]]

## $Params  
## [1] "mu\_e" "mu\_l" "mu\_t" "p" "r\_l" "rho" "b"   
##   
## $Estimate  
## mu\_e mu\_l mu\_t p r\_l rho b   
## 0.04994421 0.12824889 0.01522709 0.09797706 0.47391692 0.50465331 0.17130402   
##   
## $Lower  
## mu\_e mu\_l mu\_t p r\_l rho b   
## NaN NaN NaN 0.08916441 0.42016536 NaN NaN   
##   
## $Upper  
## mu\_e mu\_l mu\_t p r\_l rho b   
## NaN NaN NaN 0.1067897 0.5276685 NaN NaN   
##   
## $`GOF value`  
## [1] -62.66109  
##   
## $`Calibration method`  
## [1] "SANN\_wSumSquareError"  
##   
## $Sigma  
## mu\_e mu\_l mu\_t p r\_l  
## mu\_e 1.000000e+00 5.146322e-05 1.084721e-04 1.191146e-05 1.670989e-04  
## mu\_l 5.146322e-05 1.000000e+00 2.386351e-04 8.995325e-05 -6.458202e-04  
## mu\_t 1.084721e-04 2.386351e-04 1.000000e+00 -3.184523e-05 2.494390e-04  
## p 1.191146e-05 8.995325e-05 -3.184523e-05 1.000000e+00 7.329606e-05  
## r\_l 1.670989e-04 -6.458202e-04 2.494390e-04 7.329606e-05 1.000000e+00  
## rho -8.694847e-05 -1.088359e-04 1.809212e-04 3.161644e-05 -1.963692e-05  
## b 7.423549e-05 1.991019e-04 -9.562742e-05 -2.271867e-05 2.068559e-04  
## rho b  
## mu\_e -8.694847e-05 7.423549e-05  
## mu\_l -1.088359e-04 1.991019e-04  
## mu\_t 1.809212e-04 -9.562742e-05  
## p 3.161644e-05 -2.271867e-05  
## r\_l -1.963692e-05 2.068559e-04  
## rho 1.000000e+00 1.043882e-04  
## b 1.043882e-04 1.000000e+00

## O: Genetic algorithm using log\_likelihood as goodness-of-fit:----  
HID\_results$Calib\_results$Directed[[7]][[1]]

## $Params  
## [1] "mu\_e" "mu\_l" "mu\_t" "p" "r\_l" "rho" "b"   
##   
## $Estimate  
## mu\_e mu\_l mu\_t p r\_l rho b   
## 0.02088454 0.24677700 0.03009050 0.07234070 0.68200889 0.05988677 0.30047502   
##   
## $Lower  
## mu\_e mu\_l mu\_t p r\_l rho   
## 0.003918738 NaN NaN 0.055681138 0.576639957 NaN   
## b   
## -0.271023200   
##   
## $Upper  
## mu\_e mu\_l mu\_t p r\_l rho b   
## 0.03785035 NaN NaN 0.08900026 0.78737783 NaN 0.87197325   
##   
## $`GOF value`  
## [1] -1038.601  
##   
## $`Calibration method`  
## [1] "GA\_log\_likelihood"  
##   
## $Sigma  
## mu\_e mu\_l mu\_t p r\_l  
## mu\_e 1.000000e+00 1.964864e-04 -8.708574e-05 -5.612566e-05 0.0004673343  
## mu\_l 1.964864e-04 1.000000e+00 2.816276e-04 1.536640e-04 -0.0067039252  
## mu\_t -8.708574e-05 2.816276e-04 1.000000e+00 9.025629e-05 0.0007468068  
## p -5.612566e-05 1.536640e-04 9.025629e-05 1.000000e+00 0.0001994951  
## r\_l 4.673343e-04 -6.703925e-03 7.468068e-04 1.994951e-04 1.0000000000  
## rho 8.577603e-05 -3.831386e-05 2.377461e-04 -3.707902e-05 0.0003265334  
## b 2.735416e-04 -4.410726e-04 1.008052e-03 -1.338877e-04 0.0007744174  
## rho b  
## mu\_e 8.577603e-05 0.0002735416  
## mu\_l -3.831386e-05 -0.0004410726  
## mu\_t 2.377461e-04 0.0010080519  
## p -3.707902e-05 -0.0001338877  
## r\_l 3.265334e-04 0.0007744174  
## rho 1.000000e+00 -0.0010552896  
## b -1.055290e-03 1.0000000000

## O: Genetic algorithm using sum-squared-errors as goodness-of-fit:----  
HID\_results$Calib\_results$Directed[[8]][[1]]

## $Params  
## [1] "mu\_e" "mu\_l" "mu\_t" "p" "r\_l" "rho" "b"   
##   
## $Estimate  
## mu\_e mu\_l mu\_t p r\_l rho b   
## 0.02599571 0.18430894 0.01124822 0.12370265 1.06265881 0.04584952 0.41606874   
##   
## $Lower  
## mu\_e mu\_l mu\_t p r\_l rho b   
## NaN NaN NaN 0.0613914 0.2123038 NaN -4.1337500   
##   
## $Upper  
## mu\_e mu\_l mu\_t p r\_l rho b   
## NaN NaN NaN 0.1860139 1.9130138 NaN 4.9658874   
##   
## $`GOF value`  
## [1] -1147.297  
##   
## $`Calibration method`  
## [1] "GA\_wSumSquareError"  
##   
## $Sigma  
## mu\_e mu\_l mu\_t p r\_l  
## mu\_e 1.0000000000 0.0016205622 -0.0001786471 -7.779132e-04 0.009079600  
## mu\_l 0.0016205622 1.0000000000 0.0001495728 9.422725e-04 -0.031497408  
## mu\_t -0.0001786471 0.0001495728 1.0000000000 2.699876e-04 0.002917075  
## p -0.0007779132 0.0009422725 0.0002699876 1.000000e+00 0.001766570  
## r\_l 0.0090795995 -0.0314974081 0.0029170751 1.766570e-03 1.000000000  
## rho -0.0003640041 0.0016260157 -0.0001478451 -8.335402e-05 0.009392613  
## b -0.0018572742 0.0075615713 -0.0005991804 -7.591934e-05 0.045211101  
## rho b  
## mu\_e -3.640041e-04 -1.857274e-03  
## mu\_l 1.626016e-03 7.561571e-03  
## mu\_t -1.478451e-04 -5.991804e-04  
## p -8.335402e-05 -7.591934e-05  
## r\_l 9.392613e-03 4.521110e-02  
## rho 1.000000e+00 -5.314305e-03  
## b -5.314305e-03 1.000000e+00

# Transformed scale:----  
## T: Nelder-Mead using log\_likelihood as goodness-of-fit:----  
HID2\_results$Calib\_results$Directed[[1]][[1]]

## $Params  
## [1] "mu\_e" "mu\_l" "mu\_t" "p" "r\_l" "rho" "b"   
##   
## $Estimate  
## mu\_e mu\_l mu\_t p r\_l rho   
## -1.96011253 -2.66303941 -2.80788736 -1.16874063 -0.08877385 -0.36318282   
## b   
## -0.82903822   
##   
## $Lower  
## mu\_e mu\_l mu\_t p r\_l rho b   
## -21.262791 -11.540513 -8.611455 -5.923578 -22.751965 -3.713035 -5.768751   
##   
## $Upper  
## mu\_e mu\_l mu\_t p r\_l rho b   
## 17.342566 6.214434 2.995681 3.586097 22.574417 2.986669 4.110675   
##   
## $`GOF value`  
## [1] -18.03032  
##   
## $`Calibration method`  
## [1] "Nelder-Mead\_log\_likelihood"  
##   
## $Sigma  
## mu\_e mu\_l mu\_t p r\_l rho  
## mu\_e 1.000000 -44.367330 -24.3555714 4.8932320 -34.121858 12.5215509  
## mu\_l -44.367330 1.000000 11.5014724 -1.1919523 14.794770 -5.6641062  
## mu\_t -24.355571 11.501472 1.0000000 0.7115037 22.121215 -1.4607364  
## p 4.893232 -1.191952 0.7115037 1.0000000 -6.186800 0.9660926  
## r\_l -34.121858 14.794770 22.1212145 -6.1868004 1.000000 7.6438179  
## rho 12.521551 -5.664106 -1.4607364 0.9660926 7.643818 1.0000000  
## b 21.572851 -9.723750 -3.4609695 1.8554555 4.774273 4.1807706  
## b  
## mu\_e 21.572851  
## mu\_l -9.723750  
## mu\_t -3.460970  
## p 1.855455  
## r\_l 4.774273  
## rho 4.180771  
## b 1.000000

## T: Nelder-Mead using sum-squared-errors as goodness-of-fit:----  
HID2\_results$Calib\_results$Directed[[2]][[1]]

## $Params  
## [1] "mu\_e" "mu\_l" "mu\_t" "p" "r\_l" "rho" "b"   
##   
## $Estimate  
## mu\_e mu\_l mu\_t p r\_l rho b   
## -5.5064013 -2.2618051 -2.6868449 -0.6863481 -0.9537171 -0.5410032 -1.0536961   
##   
## $Lower  
## mu\_e mu\_l mu\_t p r\_l rho b   
## NaN -6.369840 -4.566053 -19.697293 -16.365941 -1.577428 -3.362708   
##   
## $Upper  
## mu\_e mu\_l mu\_t p r\_l rho b   
## NaN 1.8462294 -0.8076373 18.3245969 14.4585066 0.4954217 1.2553155   
##   
## $`GOF value`  
## [1] -1.076053e-05  
##   
## $`Calibration method`  
## [1] "Nelder-Mead\_wSumSquareError"  
##   
## $Sigma  
## mu\_e mu\_l mu\_t p r\_l rho  
## mu\_e 1.000000 13.987442 5.7943564 7.330275 -1.827253 -4.9074972  
## mu\_l 13.987442 1.000000 1.5713796 -20.698923 16.386852 -1.0326696  
## mu\_t 5.794356 1.571380 1.0000000 -7.467859 6.836007 -0.2590986  
## p 7.330275 -20.698923 -7.4678590 1.000000 -74.673614 4.9609196  
## r\_l -1.827253 16.386852 6.8360066 -74.673614 1.000000 -3.6120684  
## rho -4.907497 -1.032670 -0.2590986 4.960920 -3.612068 1.0000000  
## b -5.564739 -2.361357 -0.6468837 11.060833 -8.236772 0.6221876  
## b  
## mu\_e -5.5647394  
## mu\_l -2.3613566  
## mu\_t -0.6468837  
## p 11.0608328  
## r\_l -8.2367717  
## rho 0.6221876  
## b 1.0000000

## T: Gradient-based using log\_likelihood as goodness-of-fit:----  
HID2\_results$Calib\_results$Directed[[3]][[1]]

## $Params  
## [1] "mu\_e" "mu\_l" "mu\_t" "p" "r\_l" "rho" "b"   
##   
## $Estimate  
## mu\_e mu\_l mu\_t p r\_l rho b   
## -2.3392065 -2.4875004 -2.7504118 -0.5201908 -1.0217515 -0.4422062 -0.9378333   
##   
## $Lower  
## mu\_e mu\_l mu\_t p r\_l rho b   
## -166.47572 -29.58391 NaN NaN NaN -16.67153 -19.40975   
##   
## $Upper  
## mu\_e mu\_l mu\_t p r\_l rho b   
## 161.79731 24.60891 NaN NaN NaN 15.78712 17.53408   
##   
## $`GOF value`  
## [1] -18.02992  
##   
## $`Calibration method`  
## [1] "BFGS\_log\_likelihood"  
##   
## $Sigma  
## mu\_e mu\_l mu\_t p r\_l rho  
## mu\_e 1.0000 -1186.54226 254.62982 5887.9503 -1514.5674 1011.24409  
## mu\_l -1186.5423 1.00000 -90.43412 -1430.6178 396.9042 -198.42429  
## mu\_t 254.6298 -90.43412 1.00000 -1917.0207 636.8764 -97.06778  
## p 5887.9503 -1430.61785 -1917.02074 1.0000 5057.7466 -380.81911  
## r\_l -1514.5674 396.90417 636.87644 5057.7466 1.0000 180.61370  
## rho 1011.2441 -198.42429 -97.06778 -380.8191 180.6137 1.00000  
## b 1028.7138 -198.60655 -82.58179 -241.4165 137.1281 79.01633  
## b  
## mu\_e 1028.71384  
## mu\_l -198.60655  
## mu\_t -82.58179  
## p -241.41654  
## r\_l 137.12809  
## rho 79.01633  
## b 1.00000

## T: Gradient-based using sum-squared-errors as goodness-of-fit:----  
HID2\_results$Calib\_results$Directed[[4]][[1]]

## $Params  
## [1] "mu\_e" "mu\_l" "mu\_t" "p" "r\_l" "rho" "b"   
##   
## $Estimate  
## mu\_e mu\_l mu\_t p r\_l rho b   
## -2.2473548 -2.5089611 -2.7649179 -0.6044982 -0.9846047 -0.4379409 -0.9317470   
##   
## $Lower  
## mu\_e mu\_l mu\_t p r\_l rho   
## -73.633395 -19.364743 -14.828006 -101.338806 -44.819045 -8.795905   
## b   
## -9.680537   
##   
## $Upper  
## mu\_e mu\_l mu\_t p r\_l rho b   
## 69.138685 14.346821 9.298170 100.129809 42.849836 7.920023 7.817043   
##   
## $`GOF value`  
## [1] -5.097443e-08  
##   
## $`Calibration method`  
## [1] "BFGS\_wSumSquareError"  
##   
## $Sigma  
## mu\_e mu\_l mu\_t p r\_l rho  
## mu\_e 1.0000 -302.74714 -112.264389 -229.93535 184.72795 126.501831  
## mu\_l -302.7471 1.00000 37.308951 164.84232 -90.04066 -23.436326  
## mu\_t -112.2644 37.30895 1.000000 290.08884 -130.31099 2.466316  
## p -229.9353 164.84232 290.088839 1.00000 -1140.63296 103.891675  
## r\_l 184.7279 -90.04066 -130.310985 -1140.63296 1.00000 -35.731144  
## rho 126.5018 -23.43633 2.466316 103.89168 -35.73114 1.000000  
## b 148.2641 -29.88064 -2.850609 65.88928 -17.88984 18.632761  
## b  
## mu\_e 148.264146  
## mu\_l -29.880639  
## mu\_t -2.850609  
## p 65.889280  
## r\_l -17.889839  
## rho 18.632761  
## b 1.000000

## T: Simulated annealing using log\_likelihood as goodness-of-fit:----  
HID2\_results$Calib\_results$Directed[[5]][[1]]

## $Params  
## [1] "mu\_e" "mu\_l" "mu\_t" "p" "r\_l" "rho" "b"   
##   
## $Estimate  
## mu\_e mu\_l mu\_t p r\_l rho   
## -3.30254954 -1.02468569 -3.70762668 -2.41893614 -1.03291034 -0.91973805   
## b   
## -0.07514663   
##   
## $Lower  
## mu\_e mu\_l mu\_t p r\_l rho b   
## -7.304484 -3.534885 NaN -5.826280 -2.745507 NaN -5.047546   
##   
## $Upper  
## mu\_e mu\_l mu\_t p r\_l rho b   
## 0.6993848 1.4855137 NaN 0.9884079 0.6796860 NaN 4.8972532   
##   
## $`GOF value`  
## [1] -89.10567  
##   
## $`Calibration method`  
## [1] "SANN\_log\_likelihood"  
##   
## $Sigma  
## mu\_e mu\_l mu\_t p r\_l rho  
## mu\_e 1.0000000 2.55814052 0.73641810 3.5761804 -1.6867016 -0.13735489  
## mu\_l 2.5581405 1.00000000 0.48175855 2.1344985 -0.8980913 -0.08188251  
## mu\_t 0.7364181 0.48175855 1.00000000 0.6039726 -0.3378441 -0.02931658  
## p 3.5761804 2.13449853 0.60397263 1.0000000 -1.4409058 -0.11567756  
## r\_l -1.6867016 -0.89809134 -0.33784408 -1.4409058 1.0000000 0.06294430  
## rho -0.1373549 -0.08188251 -0.02931658 -0.1156776 0.0629443 1.00000000  
## b 5.2123823 3.17572882 0.91959448 4.4204418 -2.0988609 -0.15252732  
## b  
## mu\_e 5.2123823  
## mu\_l 3.1757288  
## mu\_t 0.9195945  
## p 4.4204418  
## r\_l -2.0988609  
## rho -0.1525273  
## b 1.0000000

## T: Simulated annealing using sum-squared-errors as goodness-of-fit:----  
HID2\_results$Calib\_results$Directed[[6]][[1]]

## $Params  
## [1] "mu\_e" "mu\_l" "mu\_t" "p" "r\_l" "rho" "b"   
##   
## $Estimate  
## mu\_e mu\_l mu\_t p r\_l rho   
## -3.50711606 -1.61154435 -3.93226930 -0.04117327 -1.80333780 -0.56171227   
## b   
## -2.68525029   
##   
## $Lower  
## mu\_e mu\_l mu\_t p r\_l rho b   
## -3.9596531 -1.7893558 -4.1773256 NaN -1.9732011 -0.6627394 NaN   
##   
## $Upper  
## mu\_e mu\_l mu\_t p r\_l rho b   
## -3.0545791 -1.4337329 -3.6872130 NaN -1.6334745 -0.4606852 NaN   
##   
## $`GOF value`  
## [1] -709.8204  
##   
## $`Calibration method`  
## [1] "SANN\_wSumSquareError"  
##   
## $Sigma  
## mu\_e mu\_l mu\_t p r\_l  
## mu\_e 1.000000000 0.010296781 -0.001103418 -0.012110227 0.009284786  
## mu\_l 0.010296781 1.000000000 -0.001963857 0.010731554 0.003552308  
## mu\_t -0.001103418 -0.001963857 1.000000000 -0.001058267 -0.002344161  
## p -0.012110227 0.010731554 -0.001058267 1.000000000 0.009251296  
## r\_l 0.009284786 0.003552308 -0.002344161 0.009251296 1.000000000  
## rho 0.002332044 0.003773952 -0.002260248 0.001093104 0.003486348  
## b 0.003228489 0.003161800 0.003772372 0.001383169 -0.001631250  
## rho b  
## mu\_e 0.0023320435 0.0032284887  
## mu\_l 0.0037739520 0.0031617999  
## mu\_t -0.0022602477 0.0037723722  
## p 0.0010931037 0.0013831691  
## r\_l 0.0034863482 -0.0016312504  
## rho 1.0000000000 -0.0004206129  
## b -0.0004206129 1.0000000000

## T: Genetic algorithm using log\_likelihood as goodness-of-fit:----  
HID2\_results$Calib\_results$Directed[[7]][[1]]

## $Params  
## [1] "mu\_e" "mu\_l" "mu\_t" "p" "r\_l" "rho" "b"   
##   
## $Estimate  
## mu\_e mu\_l mu\_t p r\_l rho b   
## -3.6870585 -1.9003687 -3.7579875 -3.2305260 -0.1788052 -2.9318252 -0.1245101   
##   
## $Lower  
## [1] NA  
##   
## $Upper  
## [1] NA  
##   
## $`GOF value`  
## [1] -1030.904  
##   
## $`Calibration method`  
## [1] "GA\_log\_likelihood"  
##   
## $Sigma  
## [1] NA

## T: Genetic algorithm using sum-squared-errors as goodness-of-fit:----  
HID2\_results$Calib\_results$Directed[[8]][[1]]

## $Params  
## [1] "mu\_e" "mu\_l" "mu\_t" "p" "r\_l" "rho" "b"   
##   
## $Estimate  
## mu\_e mu\_l mu\_t p r\_l rho b   
## -3.2226714 -1.5587731 -4.3866827 -2.2723744 0.1413917 -2.8362133 -2.5142253   
##   
## $Lower  
## [1] NA  
##   
## $Upper  
## [1] NA  
##   
## $`GOF value`  
## [1] -1148.195  
##   
## $`Calibration method`  
## [1] "GA\_wSumSquareError"  
##   
## $Sigma  
## [1] NA

## Bayesian methods:

# Sample values for the SIR methods:----  
HID\_results$Prior\_samples[['LHS\_Bayesian']] <- sample\_prior\_LHS(  
 .n\_samples = 10000,  
 .l\_params = HID\_data$l\_params)  
# SIR:----  
HID\_results$Calib\_results$Bayesian[[1]] = calibrateModel\_beyesian(  
 .b\_method = 'SIR', .func = HID\_markov,  
 .args = NULL,  
 .l\_targets = HID\_data$l\_targets,  
 .n\_resample = 10000,  
 .l\_params = HID\_data$l\_params,  
 .samples = HID\_results$Prior\_samples$LHS\_Bayesian)  
# IMIS:----  
HID\_results$Calib\_results$Bayesian[[2]] = calibrateModel\_beyesian(  
 .b\_method = 'IMIS', .func = HID\_markov,  
 .args = NULL,  
 .l\_targets = HID\_data$l\_targets,  
 .l\_params = HID\_data$l\_params,  
 .transform = FALSE,  
 .n\_resample = 10000,  
 .IMIS\_iterations = 400,  
 .IMIS\_sample = 100)

### A quick look at a sample of the calibration results:

# Original scale:----  
## Effective sample size:----  
### O: SIR:----  
HID\_results$Calib\_results$Bayesian[[1]] %>%  
 effective\_sample\_size(bayes\_calib\_output\_list = .)

## [1] 8.53724

### O: IMIS:----  
HID\_results$Calib\_results$Bayesian[[2]] %>%  
 effective\_sample\_size(bayes\_calib\_output\_list = .)

## [1] 4762.812

## Number of unique parameter sets:----  
### O: SIR:----  
HID\_results$Calib\_results$Bayesian[[1]]$Results %>%  
 distinct() %>%  
 nrow()

## [1] 79

### O: IMIS:----  
HID\_results$Calib\_results$Bayesian[[2]]$Results %>%  
 distinct() %>%  
 nrow()

## [1] 6363

# Transformed scale:----  
## Effective sample size:----  
### T: SIR:----  
HID2\_results$Calib\_results$Bayesian[[1]] %>%  
 effective\_sample\_size(bayes\_calib\_output\_list = .)

## [1] 6.637144

### T: IMIS:----  
HID2\_results$Calib\_results$Bayesian[[2]] %>%  
 effective\_sample\_size(bayes\_calib\_output\_list = .)

## [1] 4852.014

## Number of unique parameter sets:----  
### T: SIR:----  
HID2\_results$Calib\_results$Bayesian[[1]]$Results %>%  
 distinct() %>%  
 nrow()

## [1] 49

### T: IMIS:----  
HID2\_results$Calib\_results$Bayesian[[2]]$Results %>%  
 distinct() %>%  
 nrow()

## [1] 6276

# PSA:

## Sample PSA parameter draws using directed methods results where possible:

In addition to sampling PSA parameter draws, the following function ensures all results are in proper shape for the next step. Therefore, results from other methods are passed to the same function.

# Passing calibration results from directed search:  
HID\_results$PSA\_samples[["Directed"]] <- PSA\_calib\_values(  
 .l\_calib\_res\_lists = HID\_results$Calib\_results$Directed,  
 .search\_method = 'Directed',  
 .PSA\_samples = 10000,  
 .transform\_ = FALSE,  
 .l\_params = HID\_data$l\_params)  
# Passing calibration results from random search:  
HID\_results$PSA\_samples[["Random"]] <- PSA\_calib\_values(  
 .l\_calib\_res\_lists = HID\_results$Calib\_results$Random,  
 .search\_method = 'Random',  
 .PSA\_samples = 10000,  
 .transform\_ = FALSE,  
 .l\_params = HID\_data$l\_params)  
# Passing calibration results from Bayesian methods:  
HID\_results$PSA\_samples[["Bayesian"]] <- PSA\_calib\_values(  
 .l\_calib\_res\_lists = HID\_results$Calib\_results$Bayesian,  
 .search\_method = 'Bayesian',  
 .PSA\_samples = 10000,  
 .transform\_ = FALSE,  
 .l\_params = HID\_data$l\_params)

## Run PSA:

The following function runs PSA draws from each calibration process separately.

# Run all calibration results together:  
HID\_results$PSA\_results <- run\_PSA(  
 .func\_ = HID\_markov,  
 .PSA\_calib\_values\_ = c(HID\_results$PSA\_samples$Directed,  
 HID\_results$PSA\_samples$Random,  
 HID\_results$PSA\_samples$Bayesian),  
 .args\_ = list(calibrate\_ = FALSE),  
 .PSA\_unCalib\_values\_ = NULL)

# Results:

In addition to the incremental net benefit (at £30,000), the values of incremental cost and life years gained are shown below.

# Original scale:----  
HID\_results$PSA\_summary <-   
 map\_df(  
 .x = HID\_results$PSA\_results,  
 .f = function(PSA) {  
 data\_ <- tibble(  
 'mean\_inc\_Costs' = mean(PSA$inc\_cost),  
 'mean\_inc\_LY' = mean(PSA$inc\_LY),  
 'iNMB' = (mean\_inc\_LY \* 30000) - mean\_inc\_Costs,  
 'calibration\_method' = if(nrow(PSA) == 1) paste(PSA$Label[[1]], "\_\*") else PSA$Label[[1]],  
 'goodness\_of\_fit' = PSA$Overall\_fit[[1]]  
 )  
 }   
 )  
  
HID\_results$PSA\_summary

## # A tibble: 12 x 5  
## mean\_inc\_Costs mean\_inc\_LY iNMB calibration\_method goodness\_of\_fit  
## <dbl> <dbl> <dbl> <chr> <dbl>  
## 1 -77504425. 164611. 5015830421. Nelder-Mead\_log\_likel~ -18.0   
## 2 189140. 78.3 2159420. Nelder-Mead\_wSumSquar~ -0.09  
## 3 -127039854. 186737. 5729161510. BFGS\_log\_likelihood \_\* -18.0   
## 4 -125877020. 123610. 3834189292. BFGS\_wSumSquareError ~ 0   
## 5 213890. 83.6 2295457. SANN\_log\_likelihood -47.7   
## 6 216241. 85.6 2350449. SANN\_wSumSquareError -62.7   
## 7 194627. 70.0 1906362. GA\_log\_likelihood -1039.   
## 8 192249. 68.6 1867078. GA\_wSumSquareError -1147.   
## 9 275482624. 214132. 6148471210. wSumSquareError\_LHS -2.61  
## 10 275482624. 214132. 6148471210. log\_likelihood\_LHS -18.9   
## 11 159777120. 147367. 4261218016. SIR -20.2   
## 12 123979373. 131902. 3833075165. IMIS -18.3

# Transformed scale:----  
HID2\_results$PSA\_summary <-   
 map\_df(  
 .x = HID2\_results$PSA\_results,  
 .f = function(PSA) {  
 data\_ <- tibble(  
 'mean\_inc\_Costs' = mean(PSA$inc\_cost),  
 'mean\_inc\_LY' = mean(PSA$inc\_LY),  
 'iNMB' = (mean\_inc\_LY \* 30000) - mean\_inc\_Costs,  
 'calibration\_method' = if(nrow(PSA) == 1) paste(PSA$Label[[1]], "\_\*") else PSA$Label[[1]],  
 'goodness\_of\_fit' = PSA$Overall\_fit[[1]]  
 )  
 }   
 )  
  
HID2\_results$PSA\_summary

## # A tibble: 12 x 5  
## mean\_inc\_Costs mean\_inc\_LY iNMB calibration\_method goodness\_of\_fit  
## <dbl> <dbl> <dbl> <chr> <dbl>  
## 1 -117834759. 184505. 5652985673. Nelder-Mead\_log\_like~ -18.0   
## 2 -73275884. 66288. 2061922227. Nelder-Mead\_wSumSqua~ 0   
## 3 -56975711. 86250. 2644470316. BFGS\_log\_likelihood ~ -18.0   
## 4 -59613635. 93688. 2870241071. BFGS\_wSumSquareError~ 0   
## 5 1296120075. 715407. 20166091355. SANN\_log\_likelihood ~ -89.1   
## 6 174942. 70.0 1923791. SANN\_wSumSquareError -710.   
## 7 4648825. 933. 23336388. GA\_log\_likelihood \_\* -1031.   
## 8 399280. 240. 6805666. GA\_wSumSquareError \_\* -1148.   
## 9 422703450. 335842. 9652548618. wSumSquareError\_LHS -5.06  
## 10 422703450. 335842. 9652548618. log\_likelihood\_LHS -19.6   
## 11 199989394. 180821. 5224643105. SIR -20.9   
## 12 59113211. 94795. 2784732550. IMIS -18.2