Session 2:
Application of Monte Carlo simulation and Markov Chain Monte Carlo in PBPK modeling

Nan-Hung Hsieh, PhD

Postdoc @ Texas A&M Superfund Decision Science Core

12/09/2019

TEXAS A&M UNIVERSITY
SUPERFUND
RESEARCH CENTER

Content

TEXAS A&M UNIVERSITY
SUPERFUND
RESEARCH CENTER

Uncertainty and Variability

Deterministic vs Probabilistic

Traditional - Deterministic

Choose the "specific" value (or the most conservative scenario) in the risk assessment

Is it good enough?

Deterministic vs Probabilistic

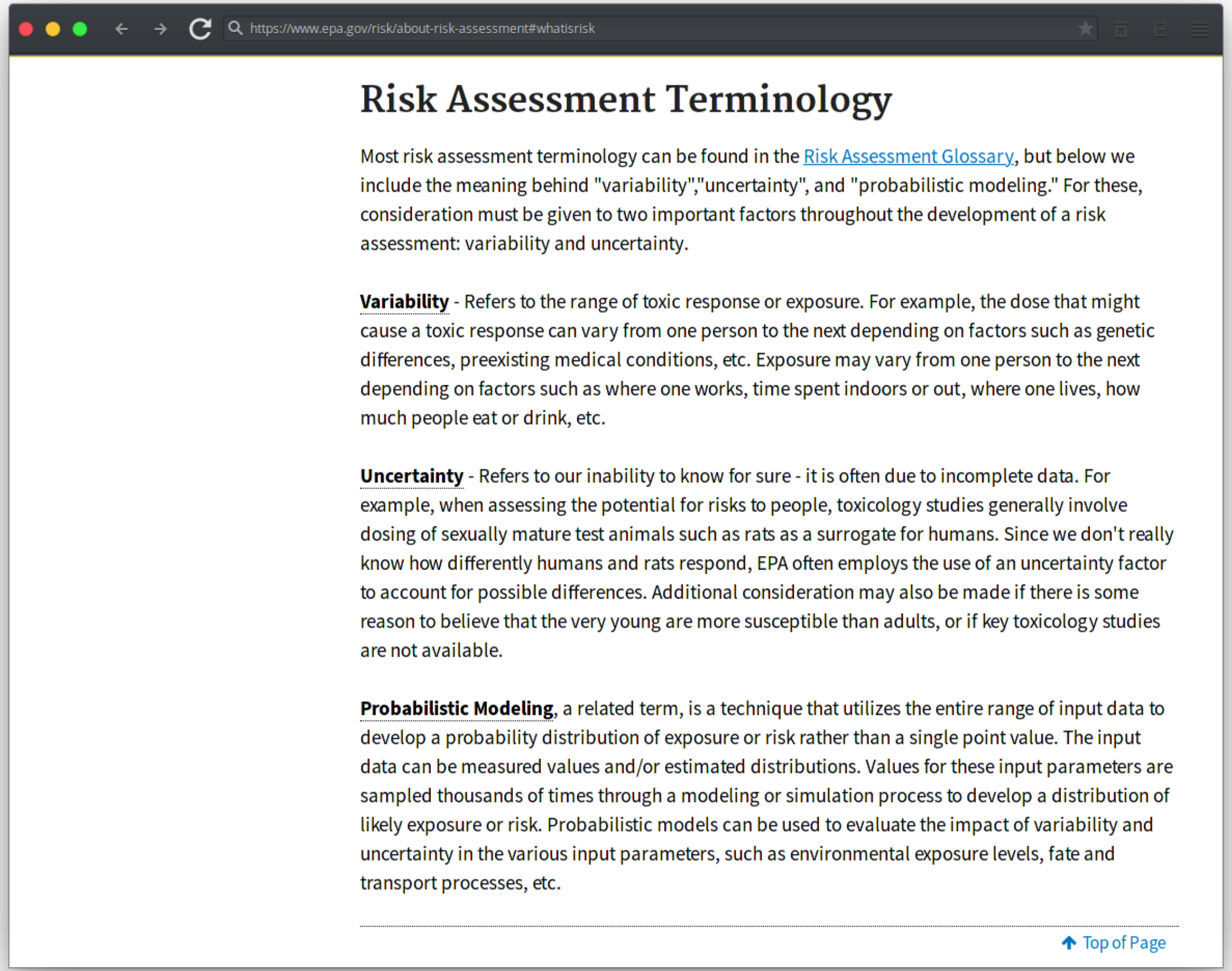
Traditional - Deterministic

Choose the "specific" value (or the most conservative scenario) in the risk assessment

Modern - Probabilistic

Combine "all" information and characterize the **uncertainty**

Modeling in Risk Assessment



https://www.epa.gov/risk/about-risk-assessment#whatisrisk

Risk Assessment Terminology

Most risk assessment terminology can be found in the Risk Assessment Glossary, but below we include the meaning behind "variability","uncertainty", and "probabilistic modeling." For these, consideration must be given to two important factors throughout the development of a risk assessment: variability and uncertainty.

Variability - Refers to the range of toxic response or exposure. For example, the dose that might cause a toxic response can vary from one person to the next depending on factors such as genetic differences, preexisting medical conditions, etc. Exposure may vary from one person to the next depending on factors such as where one works, time spent indoors or out, where one lives, how much people eat or drink, etc.

Uncertainty - Refers to our inability to know for sure - it is often due to incomplete data. For example, when assessing the potential for risks to people, toxicology studies generally involve dosing of sexually mature test animals such as rats as a surrogate for humans. Since we don't really know how differently humans and rats respond, EPA often employs the use of an uncertainty factor to account for possible differences. Additional consideration may also be made if there is some reason to believe that the very young are more susceptible than adults, or if key toxicology studies are not available.

Probabilistic Modeling, a related term, is a technique that utilizes the entire range of input data to develop a probability distribution of exposure or risk rather than a single point value. The input data can be measured values and/or estimated distributions. Values for these input parameters are sampled thousands of times through a modeling or simulation process to develop a distribution of likely exposure or risk. Probabilistic models can be used to evaluate the impact of variability and uncertainty in the various input parameters, such as environmental exposure levels, fate and transport processes, etc.

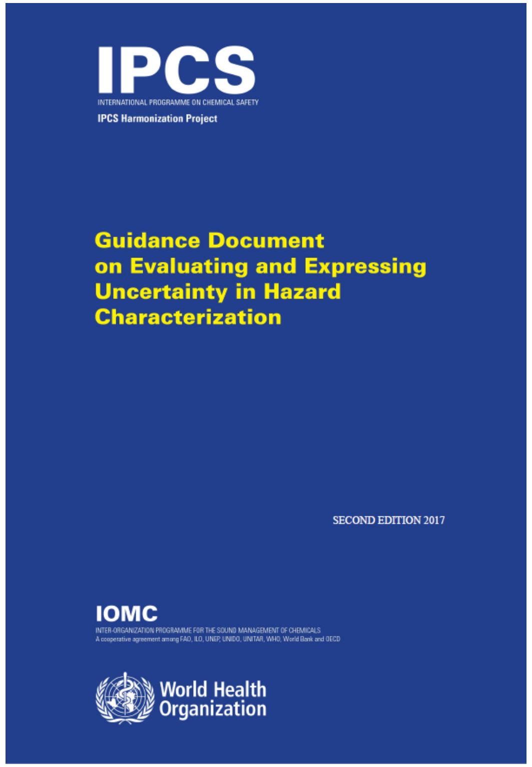
Top of Page

https://www.epa.gov/risk/about-risk-assessment#whatisrisk

Uncertainty vs. Variability

TEXAS A&M UNIVERSITY SUPERFUND RESEARCH CENTER

> **Uncertainty** relates to "lack of knowledge"" that, in theory, could be reduced by better data, whereas **variability** relates to an existing aspect of the real world that is outside our control.
>
> ***World Health Organization (2017)***


IPCS
INTERNATIONAL PROGRAMME ON CHEMICAL SAFETY
IPCS Harmonization Project

Guidance Document on Evaluating and Expressing Uncertainty in Hazard Characterization

SECOND EDITION 2017

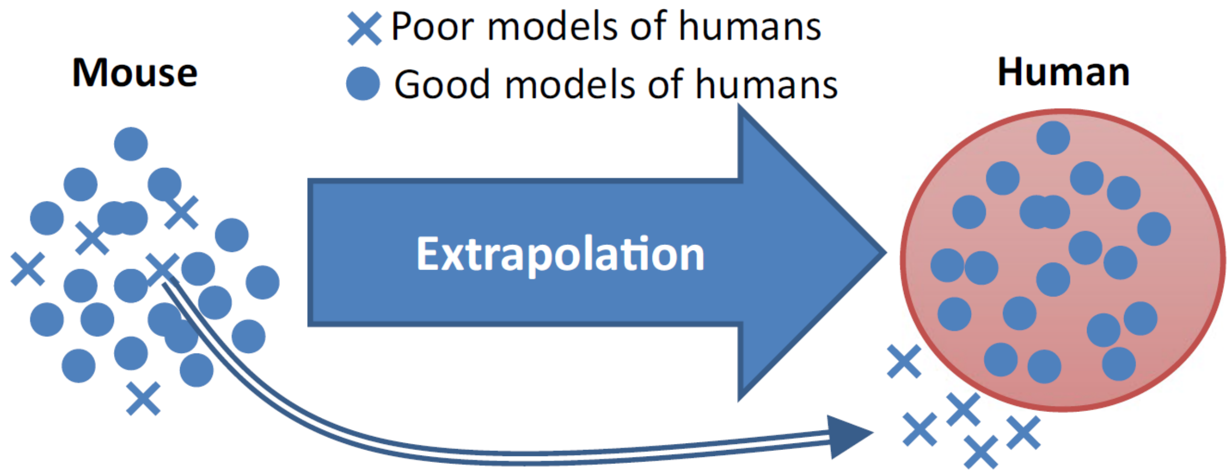
IOMC
INTER-ORGANIZATION PROGRAMME FOR THE SOUND MANAGEMENT OF CHEMICALS
A cooperative agreement among FAO, ILO, UNEP, UNIDO, UNITAR, WHO, World Bank and OECD

World Health Organization

Variability in Risk Assessment

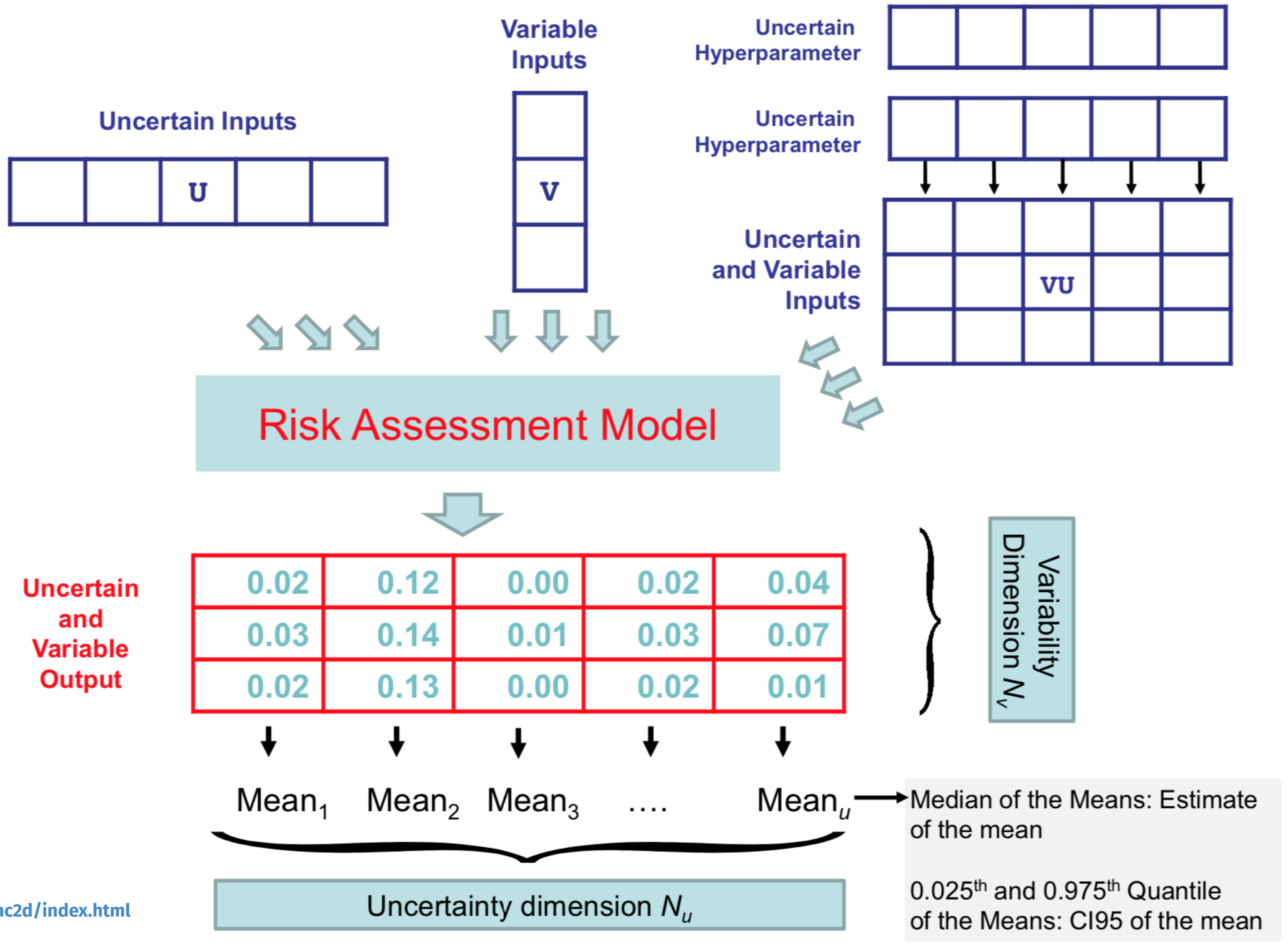
- Reduce chances using a strain that is a "poor" model of humans
- Obtaining information about "potential range" to inform risk assessment

Population Models Have Greater Likelihood of Human Relevant-Responses Than Single-Strain Models



Chiu WA and Rusyn I, 2018. Advancing chemical risk assessment decision-making with population variability data: challenges and opportunities. **https://doi.org/10.1007/s00335-017-9731-6**

Variability & Uncertainty



0.02	0.12	0.00	0.02	0.04
0.03	0.14	0.01	0.03	0.07
0.02	0.13	0.00	0.02	0.01

Median of the Means: Estimate of the mean

0.025th and 0.975th Quantile of the Means: CI95 of the mean

https://cran.r-project.org/web/packages/mc2d/index.html

--- Application ---

If you have *known* "parameters"

```
---------------------------------------->
```

Parameters / Model / Data

```
<----------------------------------------
```

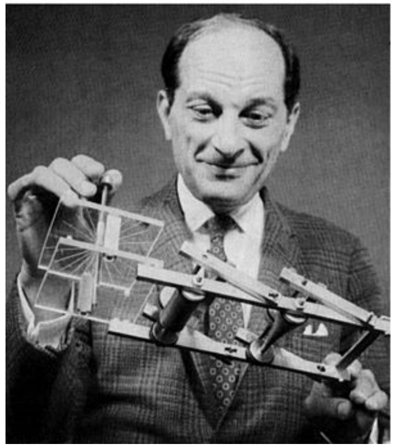
If you have *known* "data"

--- Calibration ---

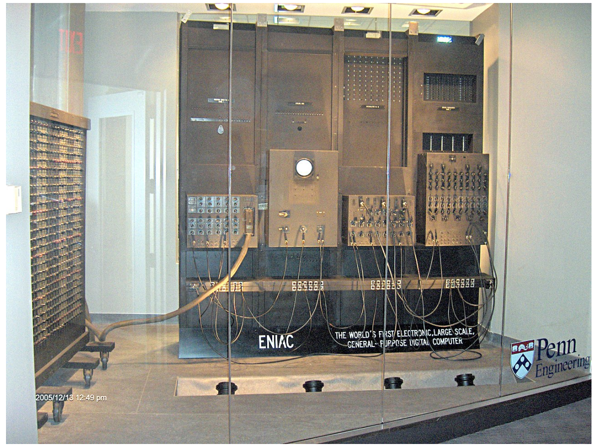
Monte Carlo Simulation

Monte Carlo Simulation

- A method of estimating the value of unknown quantity using the principle of inferential statistics
- Inferential statistics
  - **Population**: Universal information
  - **Sample**: a proper subset of population
- **Repeatedly Random Sampling**


Stanislaw Ulam

John von Neumann


ENIAC
THE WORLD'S FIRST ELECTRONIC, LARGE-SCALE, GENERAL-PURPOSE DIGITAL COMPUTER
Penn Engineering
2005/12/13 12:43 pm
ENIAC (Electronic Numerical Integrator and Computer)

Uncertainty in Risk Analysis

The objective of a **probabilistic risk analysis** is the quantification of risk from made man-made and natural activities (**Vesely and Rasmuson, 1984**).

Two major types of uncertainty need to be differentiated:

(1) Uncertainty due to physical variability

(2) Uncertainty due to lack of knowledge in

- Modeling uncertainty
- Parameter uncertainty
- Completeness uncertainty

Modeling uncertainty

Deterministic Simulation

- Define exposure unit & calculate point estimate

1-D Monte Carlo Simulation: Uncertainty

- Identify probability distributions to simulate probabilistic outputs

2-D Monte Carlo Simulation: Uncertainty & Variability

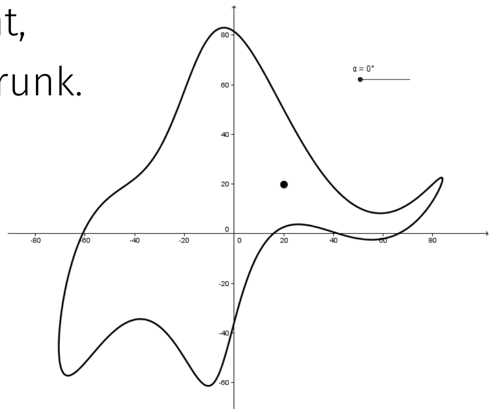
- Bayesian statistics to characterize population uncertainty and variability

Uncertainty in parameter

- **The parameter** is an element of a system that determine the model output.
- **Parameter uncertainty** comes from the model parameters that are inputs to the mathematical model but whose exact values are unknown and cannot be controlled in physical experiments.

$$y = f(x_i)$$

> With four parameters I can fit an elephant,
> and with five I can make him wiggle his trunk.
>
> -John von Neumann



Mayer J, Khairy K, Howard J. Drawing an elephant with four complex parameters. Am. J. Phys. 78, 648 (2010) **https://doi.org/10.1119/1.3254017**

Uncertainty in PBPK model parameter

Physiological parameters

Cardiac output

Blood flow rate

Tissue volume

Absorption

Absorption fraction, absorption rate, ...

Distribution

Partition coefficient, distribution fraction, ...

Metabolism

Michaelis–Menten kinetics, ...

Elimination

First order elimination rate, ...

Simulation in GNU MCSim

Monte Carlo simulations

- Perform repeated (stochastic) simulations across a randomly sampled region of the model parameter space.

Used to: Check possible simulation (under given parameter distributions) results before model calibration

SetPoints simulation

- Solves the model for a series of specified parameter sets. You can create these parameter sets yourself or use the output of a previous Monte Carlo or MCMC simulation.

Used to: Posterior predictive check, Local/global sensitivity analysis

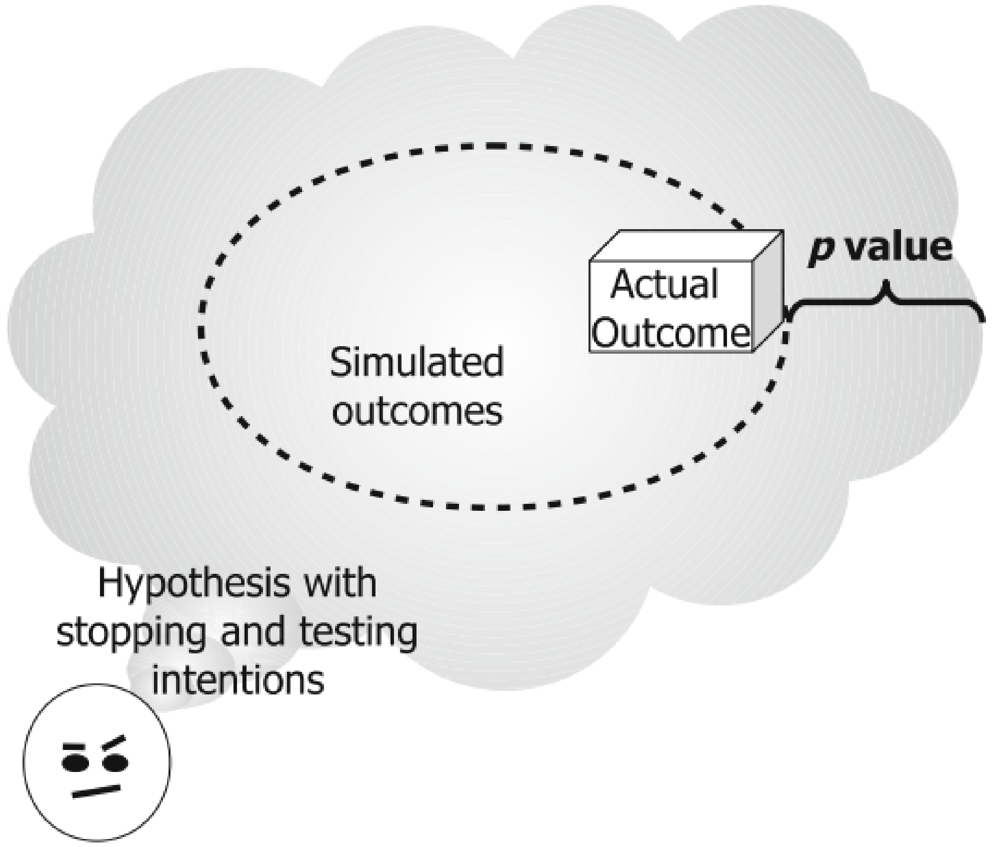
Markov Chain Monte Carlo

Currently, the Bayesian Markov chain Monte Carlo (MCMC) algorithm is an effective way to do population PBPK model calibration.

It is a powerful tool, Because...

> It gives us the opportunity to understand and quantify the "uncertainty" and "variability" from individuals to population through **data** and **model**.

Frequentist vs. Bayesian



	Frequentist	Bayesian
Hypothesis test	*p* value (null hypothesis significance test)	Bayes factor
Estimation with uncertainty	maximum likelihood estimate with confidence interval (The "New Statistics")	posterior distribution with highest density interval

https://link.springer.com/article/10.3758/s13423-016-1221-4

Bayes' rule

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

y: **Observed data**

θ: **Observed or unobserved parameter**

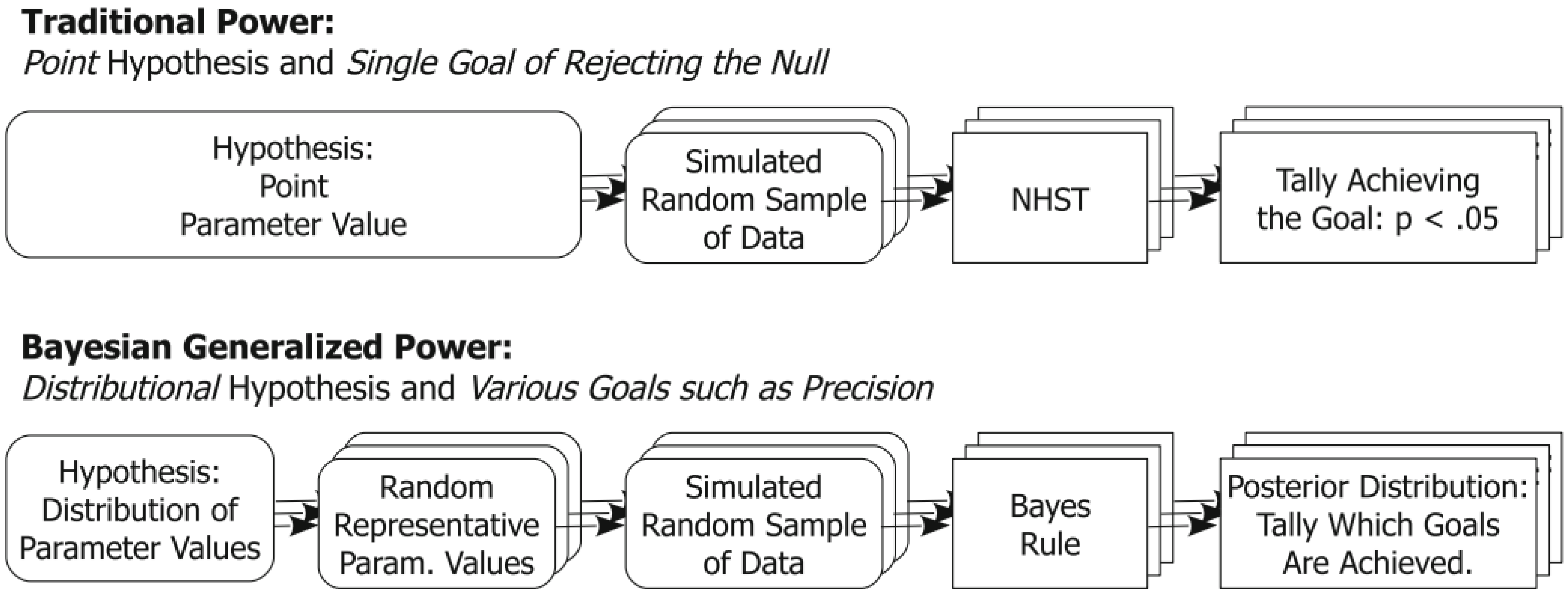
$p(\theta)$: *Prior distribution* of model parameter

$p(y|\theta)$: *Likelihood* of the experiment data given by a parameter vector

$p(\theta|y)$: *Posterior distribution*

$p(y)$: *Likelihood* of data

Frequentist vs. Bayesian

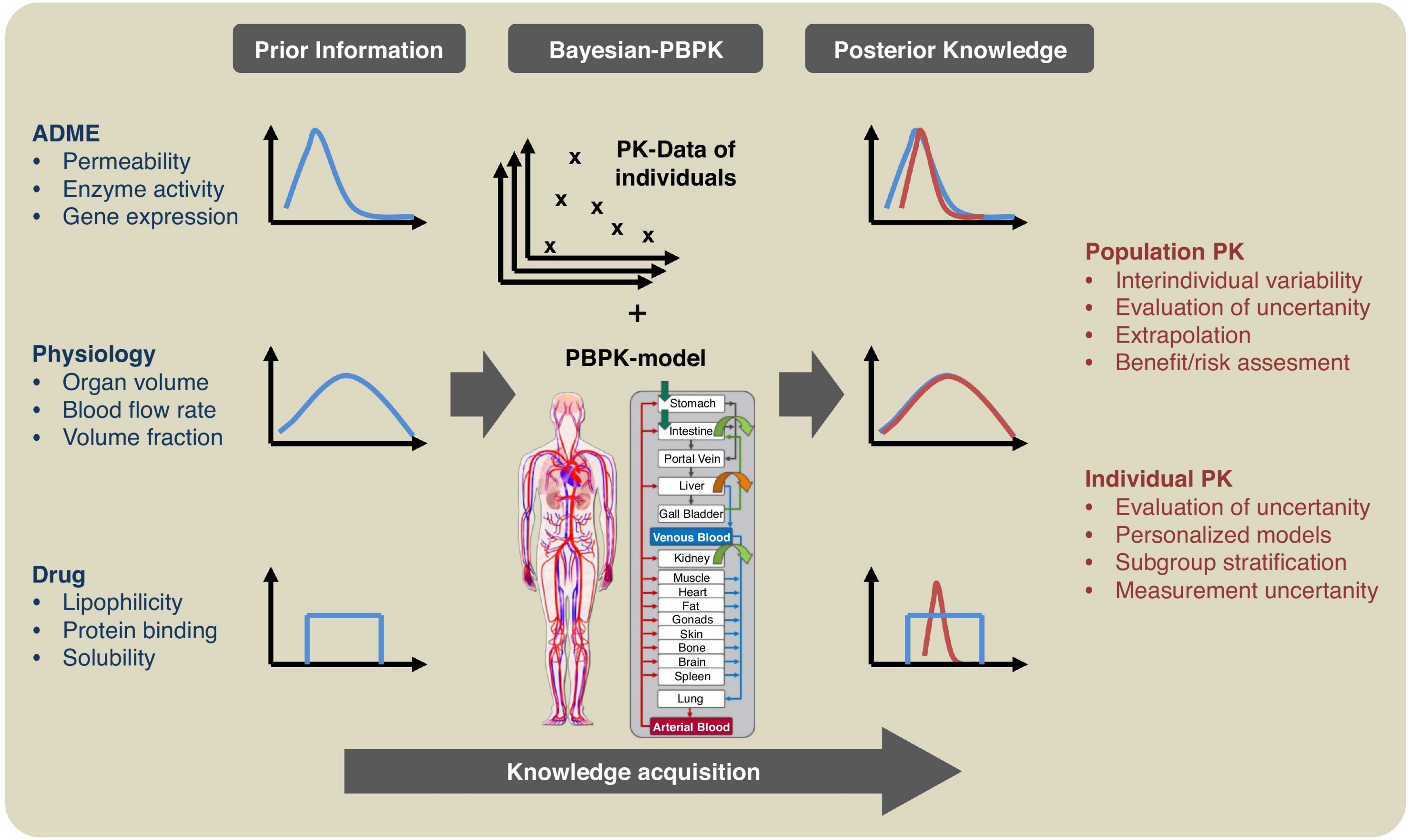

Traditional Power:
Point Hypothesis and *Single Goal of Rejecting the Null*
Hypothesis: Point Parameter Value
Simulated Random Sample of Data
NHST
Tally Achieving the Goal: p < .05
Bayesian Generalized Power:
Distributional Hypothesis and *Various Goals such as Precision*
Hypothesis: Distribution of Parameter Values
Random Representative Param. Values
Simulated Random Sample of Data
Bayes Rule
Posterior Distribution: Tally Which Goals Are Achieved.

https://link.springer.com/article/10.3758/s13423-016-1221-4

Through Markov Chain Monte Carlo ...

The product of output is not ~~best-fit~~, but "prior" and "posterior".

Probabilistic Modeling



https://doi.org/10.1016/j.ddmod.2017.08.001

Drug Discovery Today: Disease Models

Markov Chain Monte Carlo

- **Metropolis-Hastings sampling algorithm**

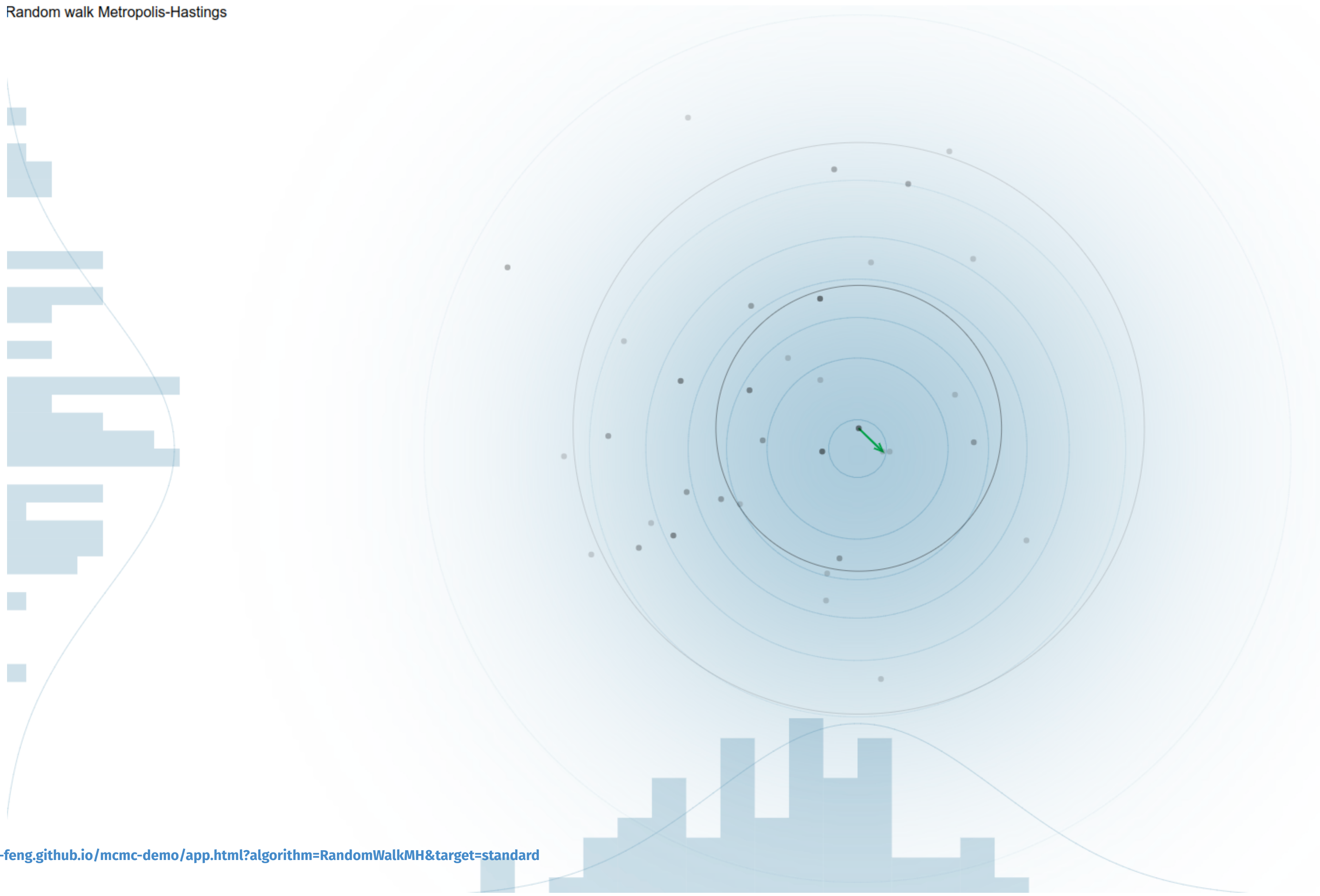
The algorithm was named for Nicholas Metropolis (physicist) and Wilfred Keith Hastings (statistician). The algorithm proceeds as follows.

Initialize

1. Pick an initial parameter sets $\theta_{t=0} = \{\theta_1, \theta_2, \ldots \theta_n\}$

Iterate

1. *Generate*: randomly generate a candidate parameter state θ' for the next sample by picking from the conditional distribution $J(\theta'|\theta_t)$
2. *Compute*: compute the acceptance probability $A\left(\theta', \theta_t\right) = \min\left(1, \frac{P(\theta')}{P(\theta_t)} \frac{J(\theta_t|\theta')}{J(\theta'|\theta_t)}\right)$
3. *Accept or Reject*:
   1. generate a uniform random number $u \in [0, 1]$
   2. if $u \leq A\left(x', x_t\right)$ accept the new state and set $\theta_{t+1} = \theta'$, otherwise reject the new state, and copy the old state forward $\theta_{t+1} = \theta_t$

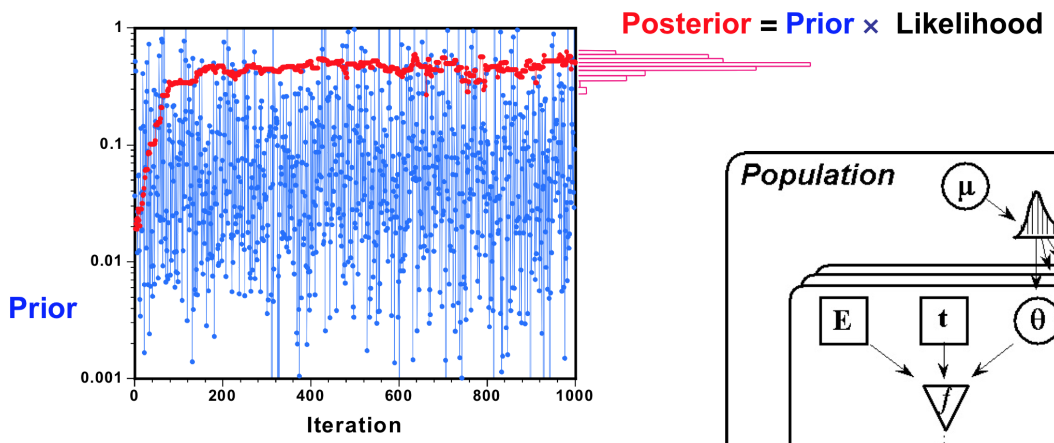


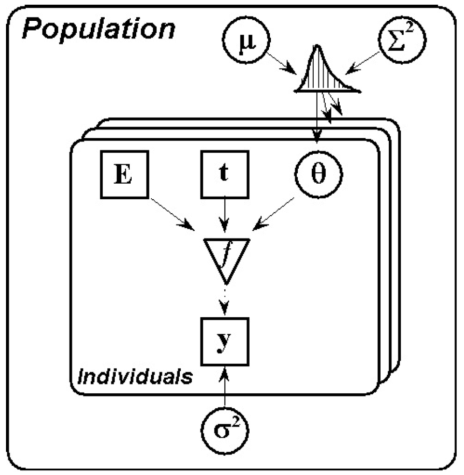
https://chi-feng.github.io/mcmc-demo/app.html?algorithm=RandomWalkMH&target=standard

Simulation in GNU MCSim

Markov-chain Monte Carlo (MCMC) simulation

- Performs a series of simulations along a Markov chain in the model parameter space.
- They can be used to obtain the Bayesian **posterior** distribution of the model parameters, given a statistical model, **prior** parameter distributions and data for which a **likelihood function** can be computed. **Used to** Model calibration





Source

Calibration & evaluation

Prepare model and input files

- Need at least 4 chains in simulation

Check convergence & graph the output result

- **Parameter**, **log-likelihood of data**
- Trace plot, density plot, correlation matrix, auto-correlation, running mean, …
- Gelman–Rubin convergence diagnostics

Evaluate the model fit

- Global evaluation
- Individual evaluation

Example - Linear model

```
## linear.model.R ####
Outputs = {y}

# Model Parameters
A = 0; #
B = 1;

CalcOutputs { y = A + B * t); }

End.
```

```
## linear_mcmc.in.R ####
MCMC ("MCMC.default.out","", # name of output
     "",               # name of data file
     2000,0,           # iterations, print predictions flag,
     1,2000,           # printing frequency, iters to print
     10101010);        # random seed (default )

Level {

  Distrib(A, Normal, 0, 2); # prior of intercept
  Distrib(B, Normal, 1, 2); # prior of slope

  Likelihood(y, Normal, Prediction(y), 0.05);

  Simulation {
    PrintStep (y, 0, 10, 1);
    Data  (y, 0.01, 0.15, 2.32, 4.33, 4.61, 6.68,
                7.89, 7.13, 7.27, 9.4, 10.0);
  }
}

End.
```

Example - MCMC simulation

```r
model <- "models/linear.model"
input <- "inputs/linear.mcmc.in"
set.seed(1111)
out <- mcsim(model, input)
```

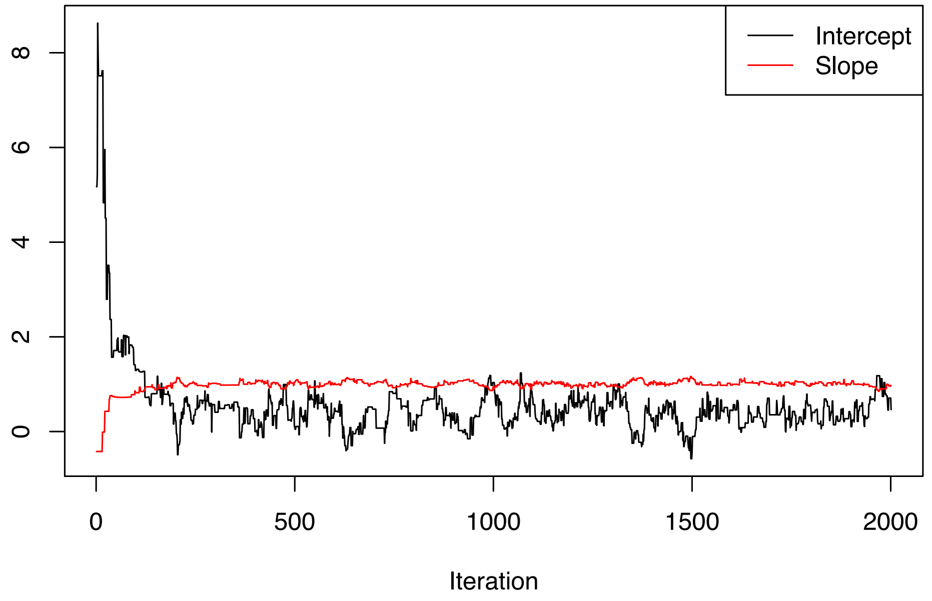
```r
head(out)
```

```
##   iter    A.1.      B.1.    LnPrior    LnData LnPosterior
## 1    0 5.17187 -0.421849  -6.820405 -591.1577   -597.9781
## 2    1 5.17187 -0.421849  -6.820405 -591.1577   -597.9781
## 3    2 5.43465 -0.421849  -7.168811 -565.2515   -572.4203
## 4    3 8.62813 -0.421849 -12.782460 -493.2532   -506.0356
## 5    4 7.97506 -0.421849 -11.427070 -471.4773   -482.9044
## 6    5 7.51347 -0.421849 -10.533410 -467.4057   -477.9391
```

```r
tail(out, 4)
```

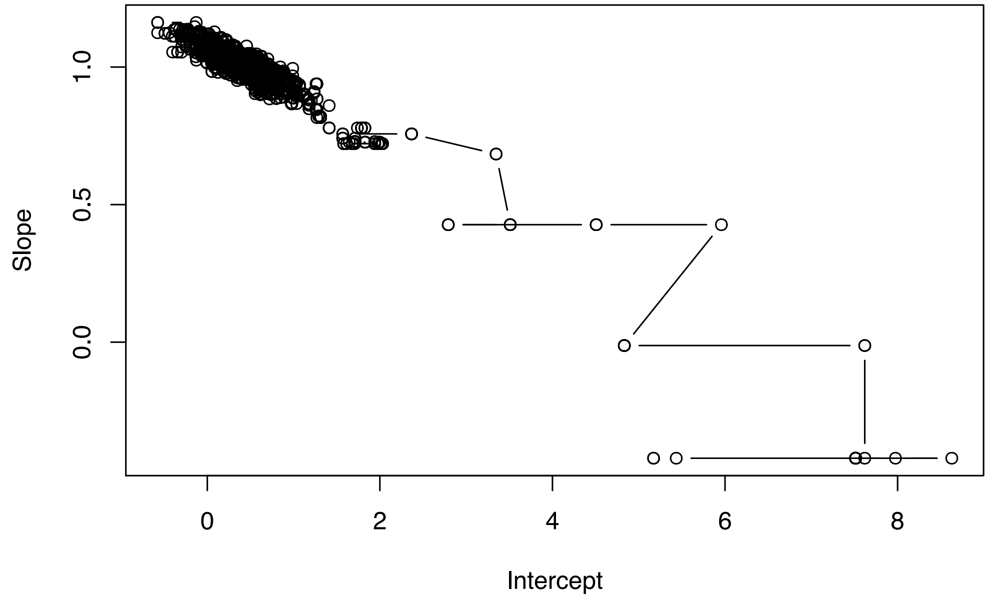
```
##      iter     A.1.     B.1.   LnPrior    LnData LnPosterior
## 1998 1997 0.706138 0.957902 -3.286722 -19.06480   -22.35153
## 1999 1998 0.706138 0.957902 -3.286722 -19.06480   -22.35153
## 2000 1999 0.706138 0.974017 -3.286585 -19.13584   -22.42242
## 2001 2000 0.462481 0.974017 -3.250992 -18.92306   -22.17405
```

```r
plot(out$A.1., type = "l", xlab = "Iteration", ylab = "")
lines(out$B.1., col = 2)
legend("topright", legend = c("Intercept", "Slope"),
       col = c(1,2), lty = 1)
```

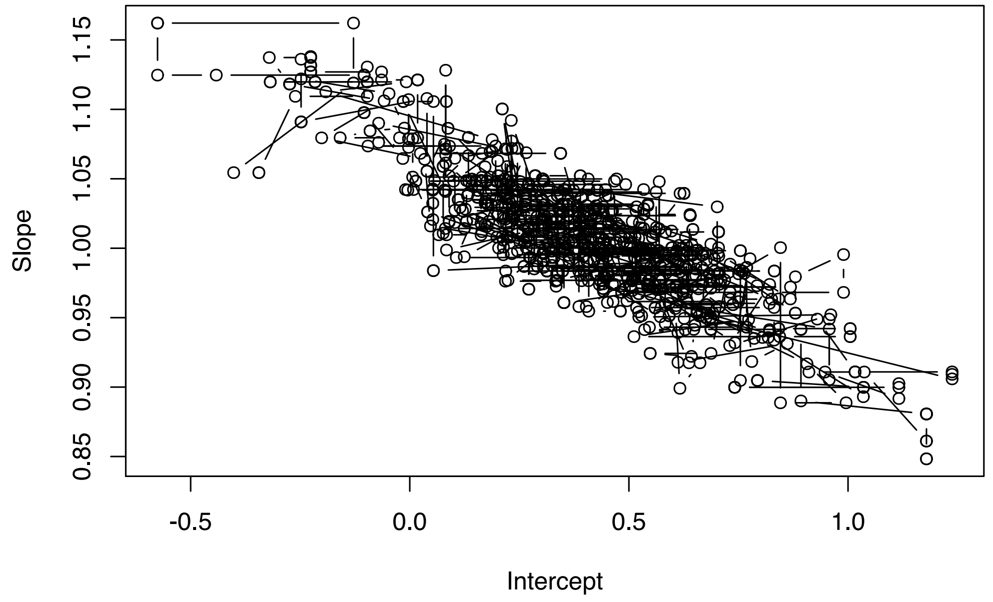


Example - Posterior check

```r
plot(out$A.1., out$B.1., type = "b",
     xlab = "Intercept", ylab = "Slope")
```



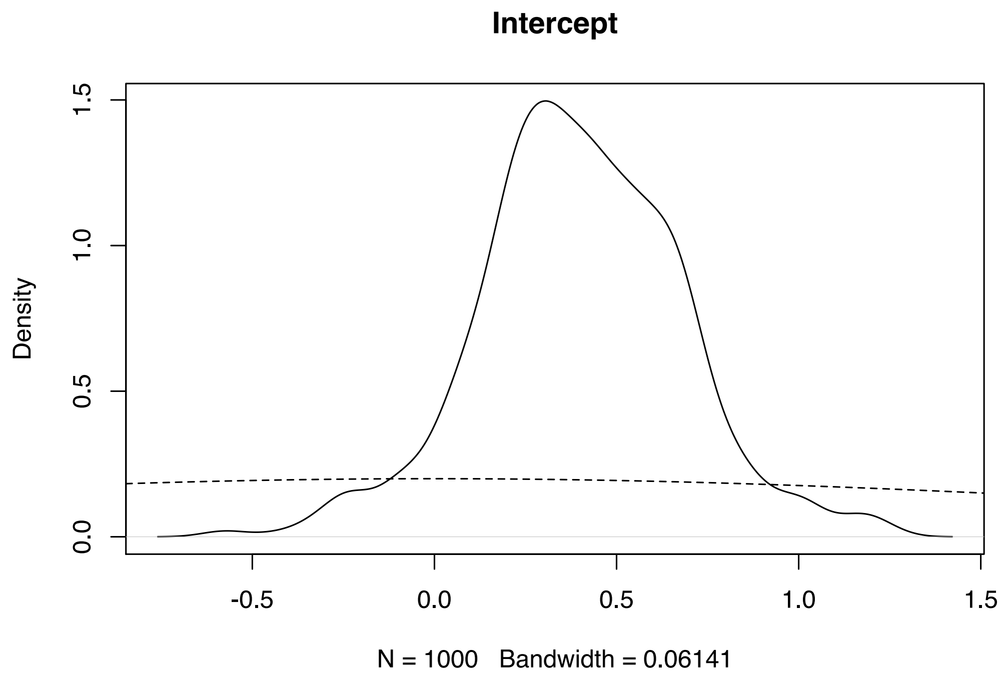
```r
str ← ceiling(nrow(out)/2) + 1
end ← nrow(out)
j ← c(str:end)
plot(out$A.1.[j], out$B.1.[j], type = "b",
     xlab = "Intercept", ylab = "Slope")
```

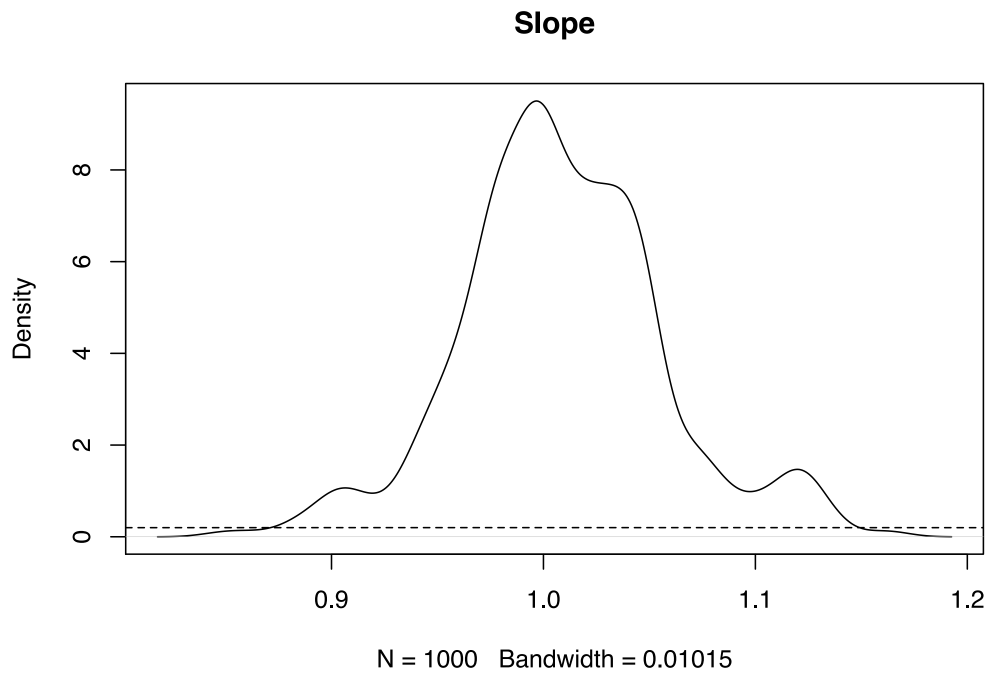


Example - Posterior check

```
out$A.1.[j] %>% density() %>% plot(main = "Intercept")
plot.rng ← seq(par("usr")[1], par("usr")[2], length.out = 1000)
prior.dens ← do.call("dnorm", c(list(x=plot.rng), c(0,2)))
lines(plot.rng, prior.dens, lty="dashed")
```

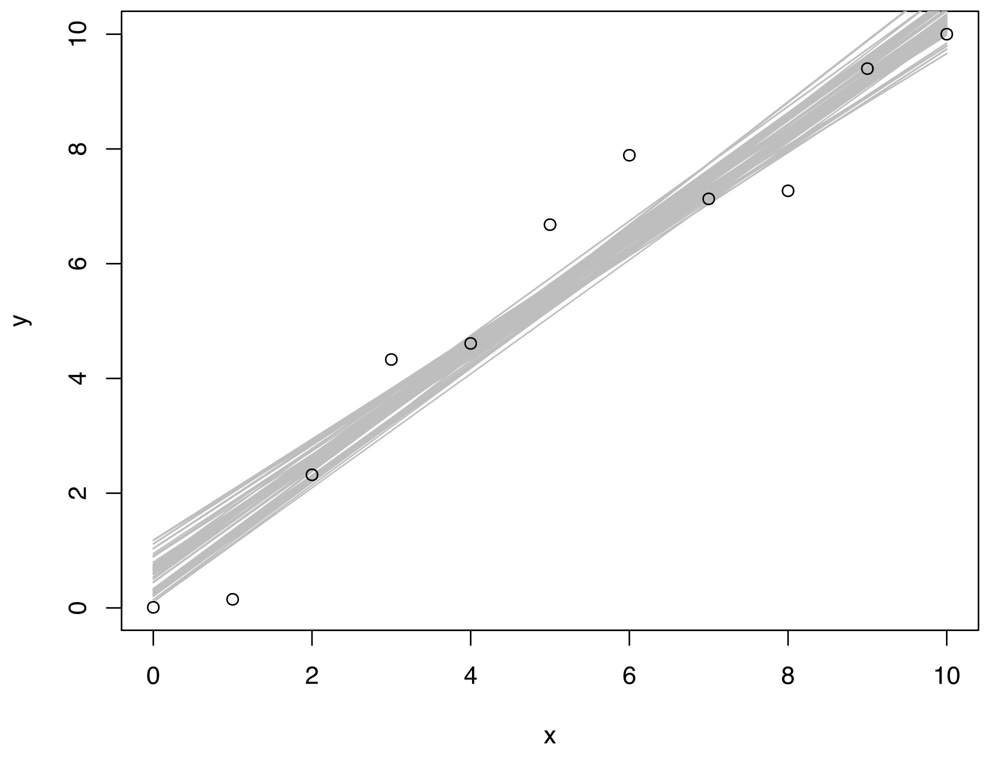
```
out$B.1.[j] %>% density() %>% plot(main = "Slope")
plot.rng ← seq(par("usr")[1], par("usr")[2], length.out = 1000)
prior.dens ← do.call("dnorm", c(list(x=plot.rng), c(1,2)))
lines(plot.rng, prior.dens, lty="dashed")
```





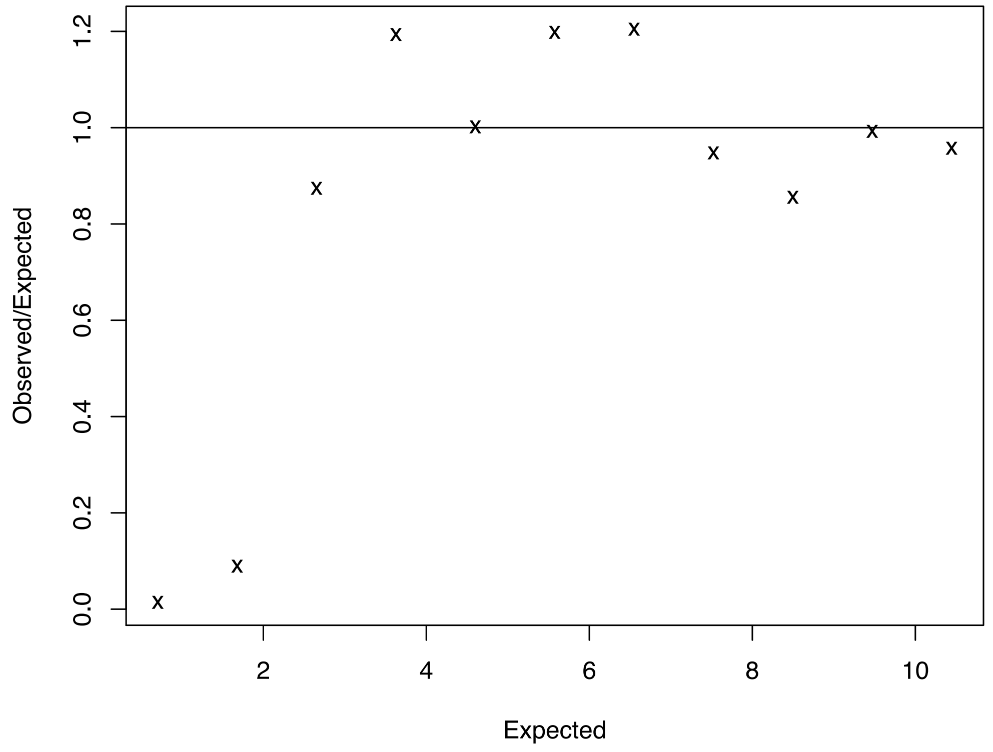
Example - Evaluation of prediction

```r
# Observed
x ← seq(0,10,1)
y ← c(0.0, 0.15, 2.32, 4.33, 4.61, 6.68,
       7.89, 7.13, 7.27, 9.4, 10.0)

# Expected
dim.x ← ncol(out)
for(i in 1:11){
  out[,ncol(out)+1] ← out$A.1. + out$B.1.*x[i]
}

# Plot
plot(x, y, pch ="")
for(i in 1901:2000){
  lines(x, out[i,c((dim.x+1):ncol(out))],col="grey")
}
points(x, y)
```



Example - Evaluation of prediction

```
# Use prediction from the last iteration
i ← 2000
Expected ← out[i, c((dim.x+1):ncol(out))] %>%
  as.numeric()
Observed ← c(0.01, 0.15, 2.32, 4.33, 4.61, 6.68,
              7.89, 7.13, 7.27, 9.4, 10.0)

# Plot
plot(Expected, Observed/Expected, pch='x')
abline(1,0)
```



General Bayesian-PBPK Workflow

1 Model constructing or translating

2 Verify modeling result

- **Compare with published result**
- **Mass balance**

3 Uncertainty (and sensitivity) analysis

4 Model calibration and validation

- **Markov chain Monte Carlo**
    - Diagnostics (Goodness-of-fit, convergence)

Summary

- In the real-word study, we need to consider the **uncertainty** (from different sources) and **variability** (inter or intra-individual data) to include all possible scenarios.

- If we have parameter, we can apply **Monte Carlo technique** to qunatify the uncertainty and variability.

- If we have data, we can calibrate the "unknown" parameter (prior) to "known" parameter (posterior) in our model through **Bayesian statistics**.

Hands on Exercise

Hands on Exercise

Task 1. **Uncertainty analysis on PK model** (`code`: **https://rpubs.com/Nanhung/SRP19_6**)

- Before model calibration, we need to learn how to conduct Monte Carlo simulation to set the proper parameter distribution

Task 2. **Model calibration** (`code`: **https://rpubs.com/Nanhung/SRP19_7**)

- After the uncertainty analysis, we can calibrate the model parameters by Markov chain Monte Carlo technique

Task 3. **Monte Carlo Simulation for PBPK model** (`code`: **https://rpubs.com/Nanhung/SRP19_8**)

- Learn how parameter effect on model variable in PBPK model

Task 4. **PBPK model in MCSim** (`code`: **https://rpubs.com/Nanhung/SRP19_9**)

- Sometimes, the simulation process in R is very computational expensive. We need to solve it.

Hands on Exercise

Task 1: **Uncertainty analysis on PK model**

- In the previous exercise, we find that the predcited result can not used to describe the real cases.
- Therefore, we need to conduct the uncertainty analysis to figure out how to reset the model parameter.

Hands on Exercise

Task 2: **Model calibration**

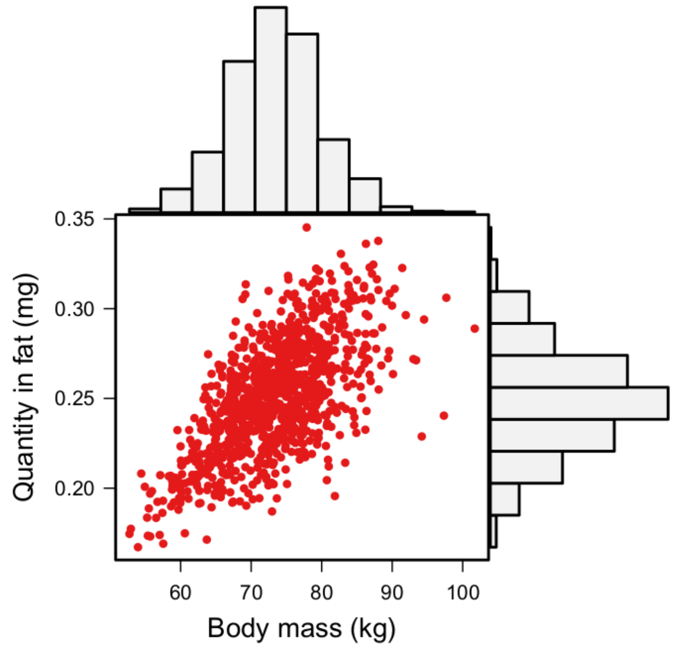
- After the uncertainty analysis, we can calibrate the model parameters by Markov chain Monte Carlo technique

- Use the parameter distributions that we test in uncertainty analysis and conduct MCMC simulation to do model calibration.

Hands on Exercise

Task 3: **Monte Carlo Simulation for PBPK model**

- Reproduce the published Monte Carlo analysis result in Bois and Brochot (2016)*
  - Testing parameter include body mass (`BDM`), pulmonary flow (`Flow_pul`), partition coefficient of arterial blood (`PC_art`) and metabolism rate (`Kmetwp`)
- Construct the relationship between body mass and quantity in fat



[*] Bois F.Y., Brochot C. (2016) **Modeling Pharmacokinetics**. In: Benfenati E. (eds) In Silico Methods for Predicting Drug Toxicity. Methods in Molecular Biology, vol 1425. Humana Press, New York, NY

Hands on Exercise

Task 4: **Monte Carlo Simulation for PBPK model (MCSim)**

- The Monte Carlo Simulation take a little bit longer with `ode` function in **deSolve** package.
- Therefore we want to improve the computational speed. Now, rewrite the R model code to **MCSim** and conduct Monte Carlo Simulation with the same parameter setting.
- The goal of this exercise is to compare the computational time and output (MCSim vs. R).

```
Distrib (BDM, Normal, 73, 7.3);
Distrib (Flow_pul, Normal, 5, 0.5);
Distrib (PC_art, Normal, 2, 0.2);
Distrib (Kmetwp, Normal, 0.25, 0.025);
```