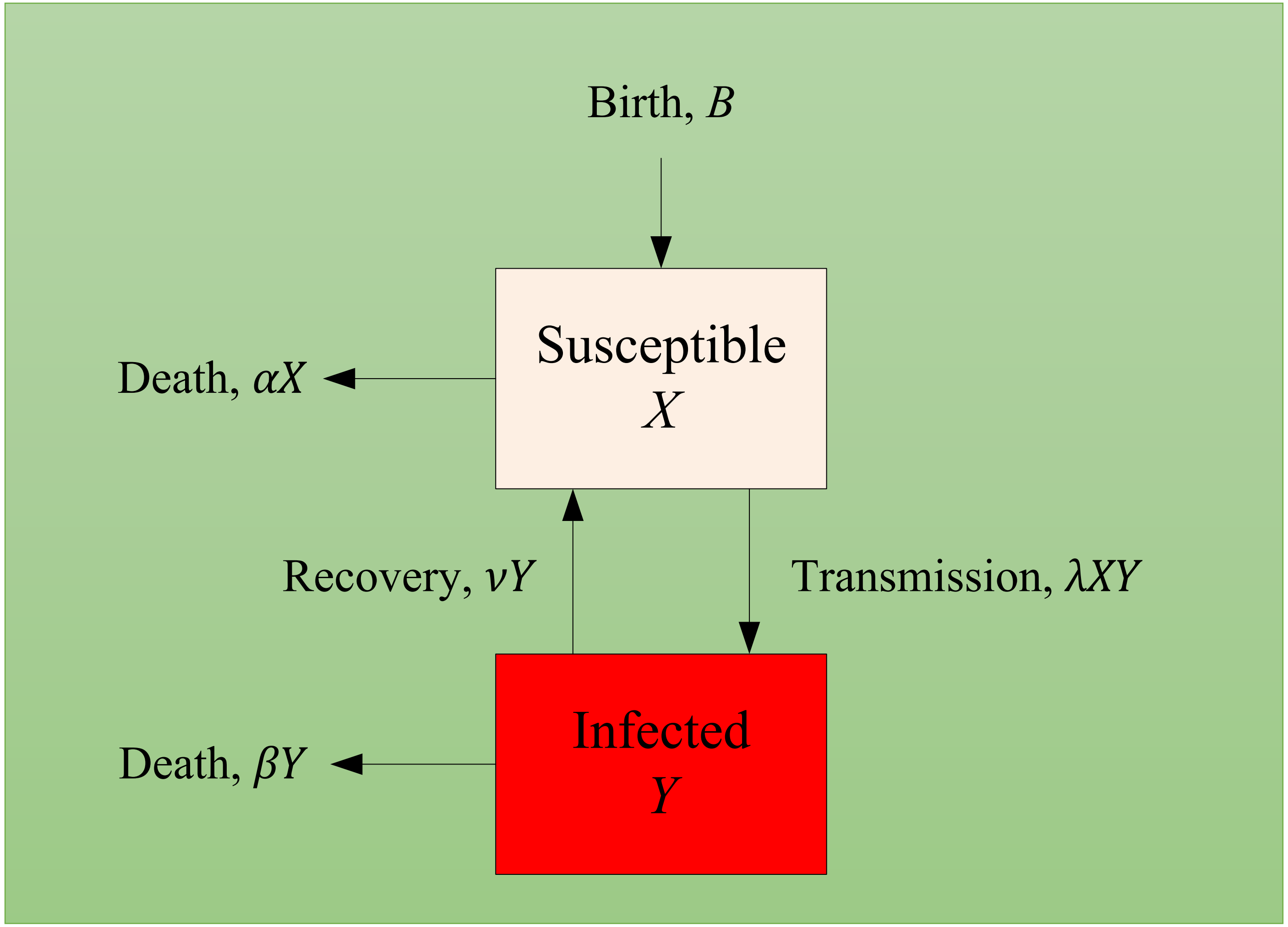
CPE 213 Data Models
(a.k.a. Data Modeling and Visualization)

Lecture 8: Simulating data distribution using Monte Carlo simulation

Asst. Prof. Dr. Santitham Prom-on

Department of Computer Engineering, Faculty of Engineering
King Mongkut's University of Technology Thonburi

Modeling virus dynamics



More realistic virus dynamic model

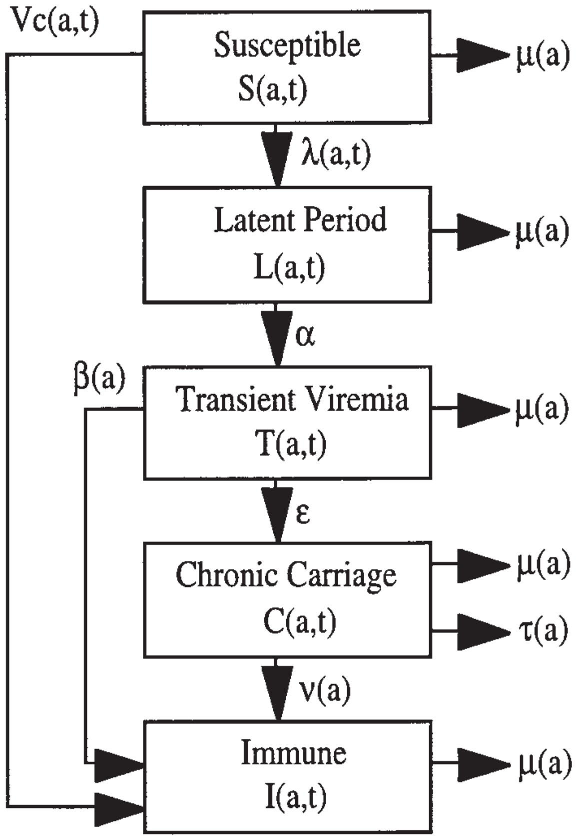


Figure 1 Flowchart of hepatitis B virus (HBV) transmission in a population

$\lambda(a,t)$: the force of HBV infection.

α: the rate of transition from latent period to temporary HBV viraemia.

$\beta(a)$: the risk of transient viraemia progressing to chronic HBV carrier state.

ε: the rate of transition from temporary HBV viraemia to immune per time unit.

$\nu(a)$: the rate of HBV clearance in chronic HBV carriers.

$\tau(a)$: the mortality rate of HBV related diseases.

$\propto(a)$: the age-specific mortality rate of non-HBV related diseases.

Vc(a,t): the effectiveness of hepatitis B vaccine immunization.

$$\lambda(a,0) = \begin{cases} 0.13074116 - 0.01362531a + 0.00046463a^2 \\ -0.00000489a^3, 0 \leqslant a \leqslant 47.5, \\ \lambda(47.5,0), a > 47.5 \end{cases}$$

$$\beta(a) = 0.706004\exp(-0.787711a) + 0.08464$$
$$\nu(a) = 0.00227005a - 0.00011211a^2 + 0.00000149a^3$$
$$\tau(a) = 1/[1 + \exp(11.80965 - 0.16887177a + 0.0007375a^2)]$$

Specific input…specific output

Inputs
e.g. $x1 = 100$
$x2 = 500$

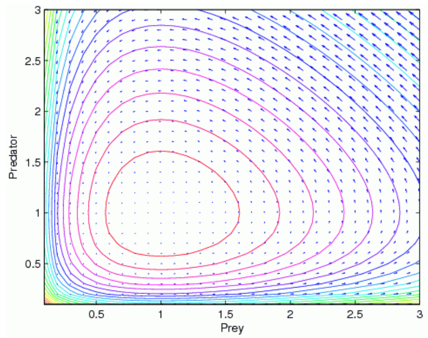
→ Model →

Output
e.g. $y = 50$

Specific input…specific behavior

Initial condition
e.g. x0 = 100
y0 = 500

Model



Simulating dynamical systems

```
set initial conditions
set t=0
do
 compute systems at t
 increase t
while t < t_max
```

Form comparison

Analytical model

- Analytic solution
- Predicting output based on
  - a set of parameters
  - initial conditions

Simulation model

- No closed form analytic solution
- Predicting output based on
  - Simulated inputs

Method comparisons

Analytical method (e.g. solving ODEs)
- Can examine many decision points at once
- But limited to simple models

Numerical method (e.g. Euler approximation)
- Can handle more complex models but still limited
- Often have to repeat computation for each decision point

Simulation modeling (e.g. Monte Carlo)
- Can handle very complex and realistic systems
- But has to be repeated for each decision point

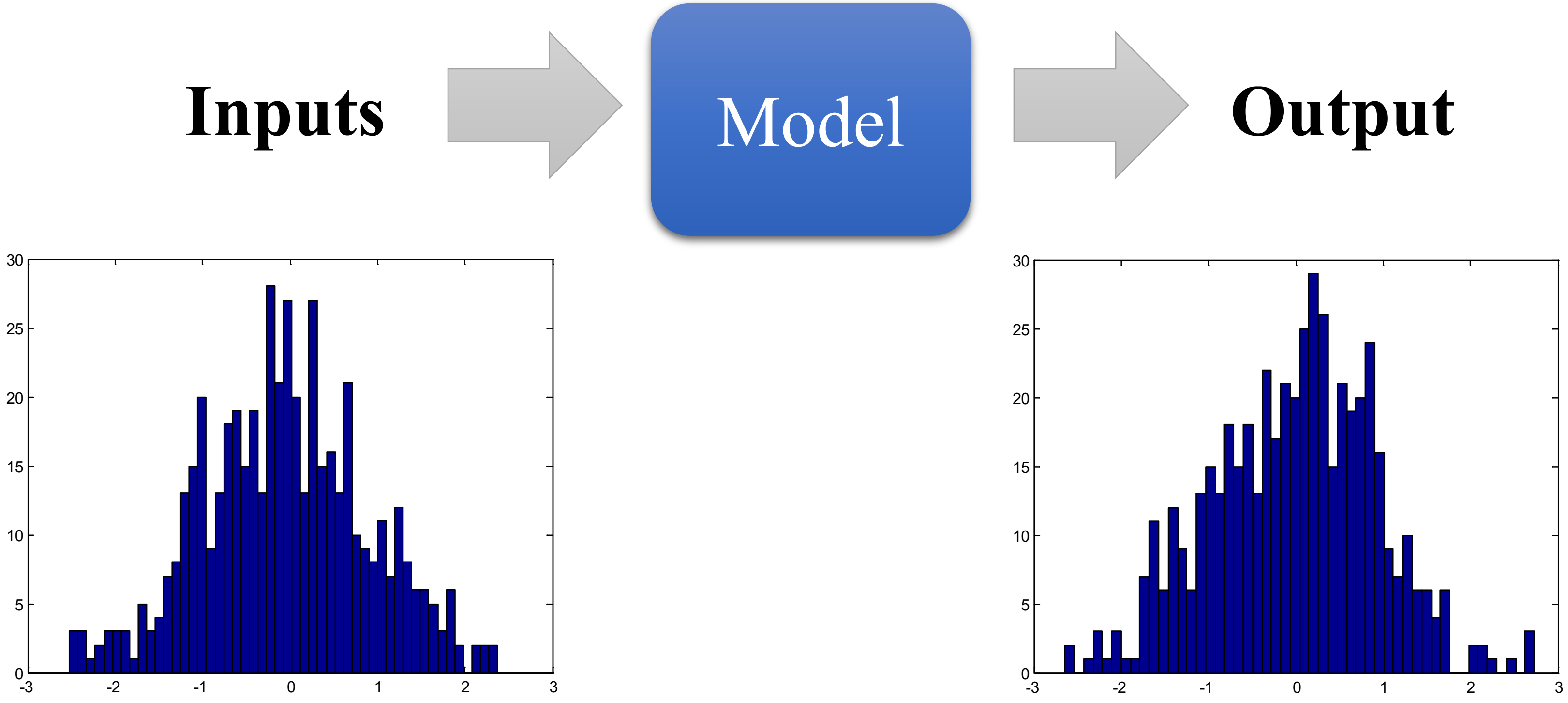
Why use simulation?

- To understand complex stochastic systems
- To control complex stochastic systems

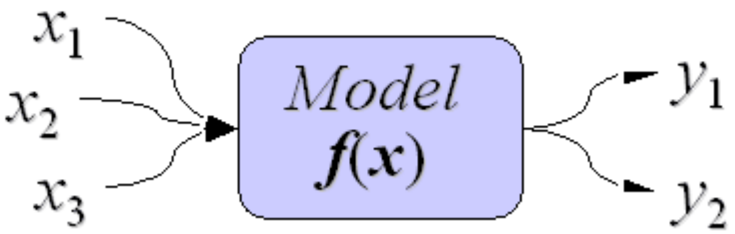
Computer Simulation

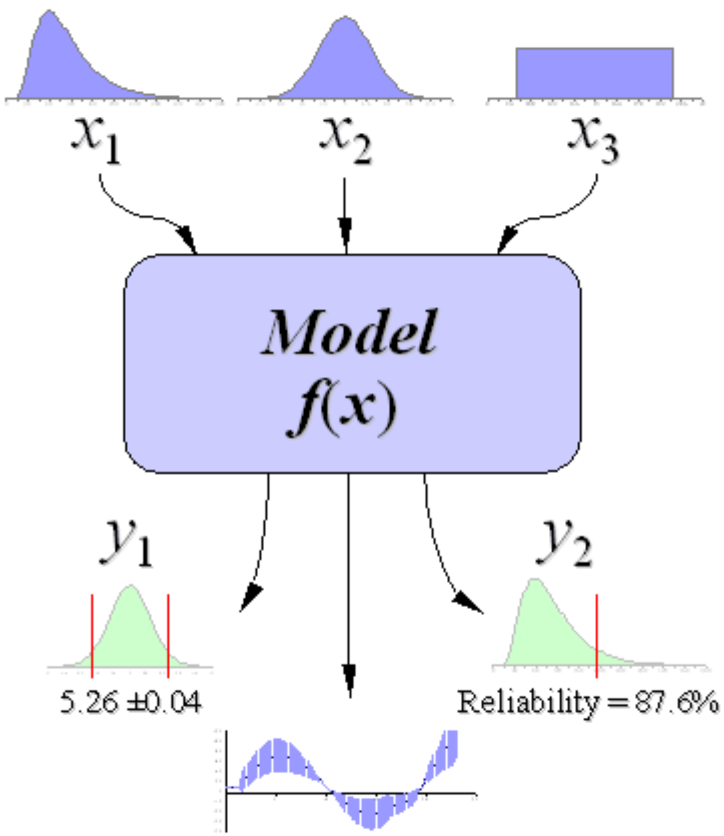
Mathematical Model + Simulated Inputs / Parameters = Computer Simulation

Simulated input…simulated output



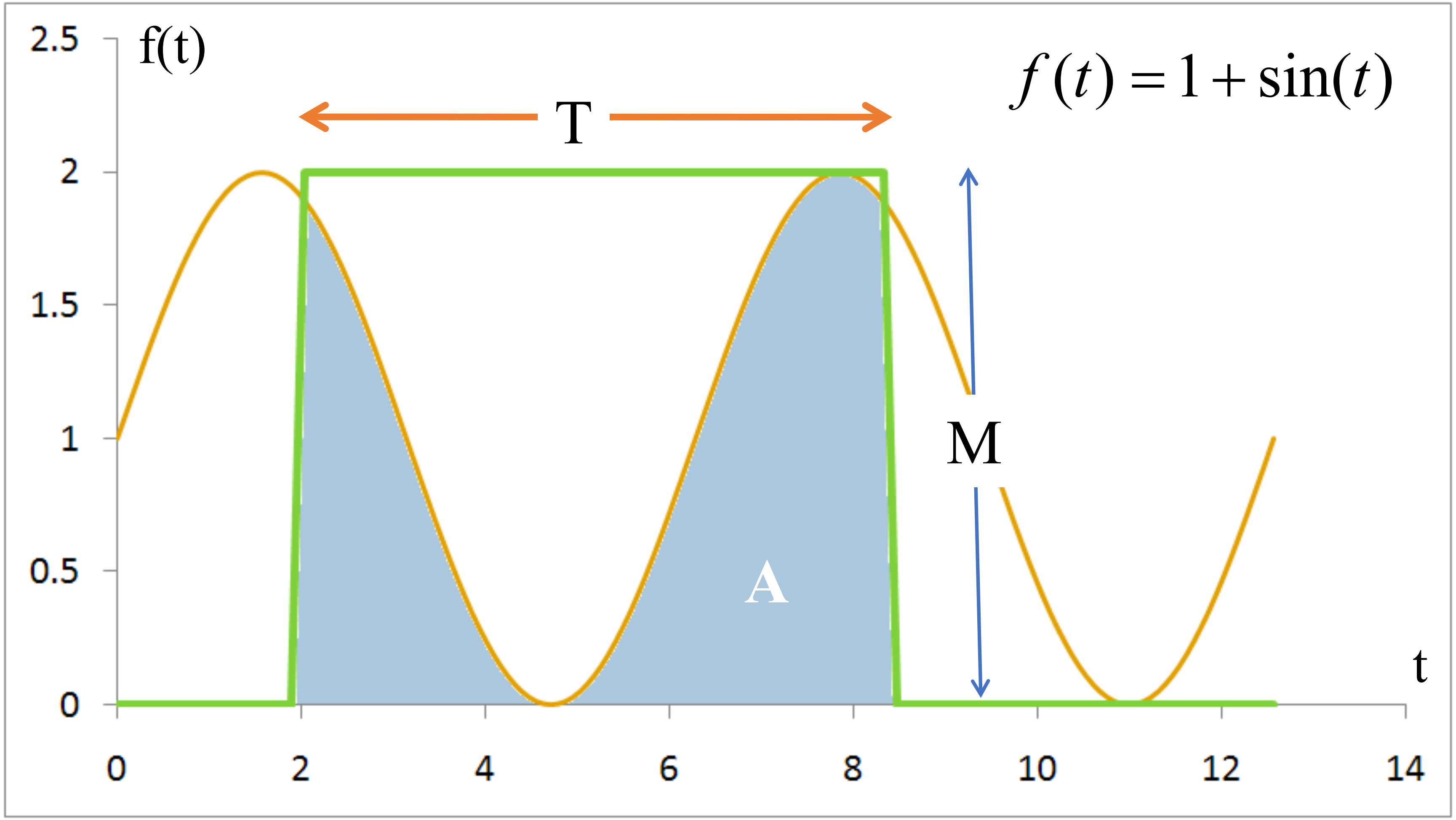
Simulating deterministic models





Integration of complex / unknown function

Example I: Area under a function
Problem statement



Analytical approach

$$A = \int_{t_0}^{t_0+T} f(t)\,dt$$

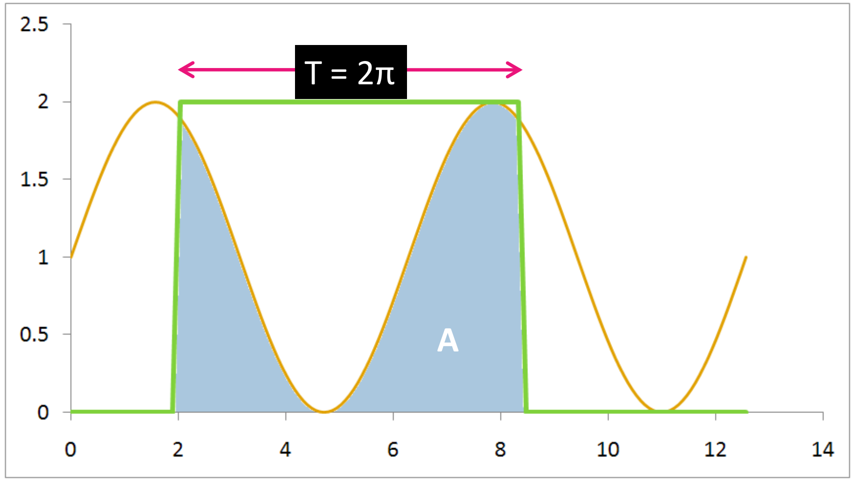
$$= \int_{t_0}^{t_0+2\pi} \left(1+\sin(t)\right)dt$$

$$= t\Big|_{t_0}^{t_0+2\pi} - \cos\left(t\Big|_{t_0}^{t_0+2\pi}\right)$$

$$= \left(t_0 + 2\pi - t_0\right) - 0$$

$$= 2\pi$$

$$= 6.2832$$



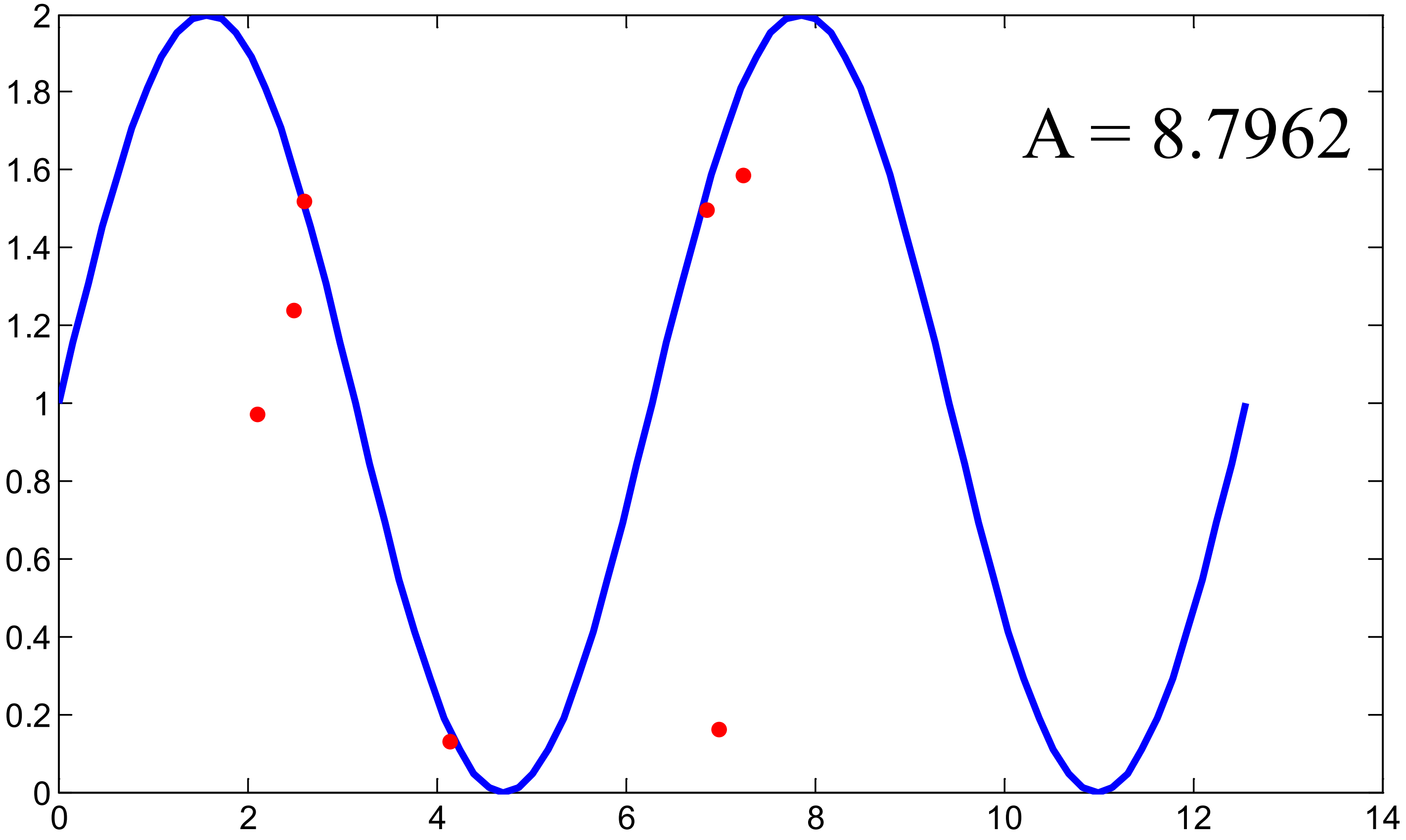
Formulate a Monte Carlo simulation model of area under a function

```
library(tidyverse)

f <- function(x) {
  return(1+sin(x))
}

run <- 1000
t <- runif(n=run, min=2, max=2+2*pi)
y <- runif(n=run, min=0, max=2)
f_t <- f(t)
print(paste0("Approx. Pi:  ",2*pi*2*sum(y < f_t)/run))
```

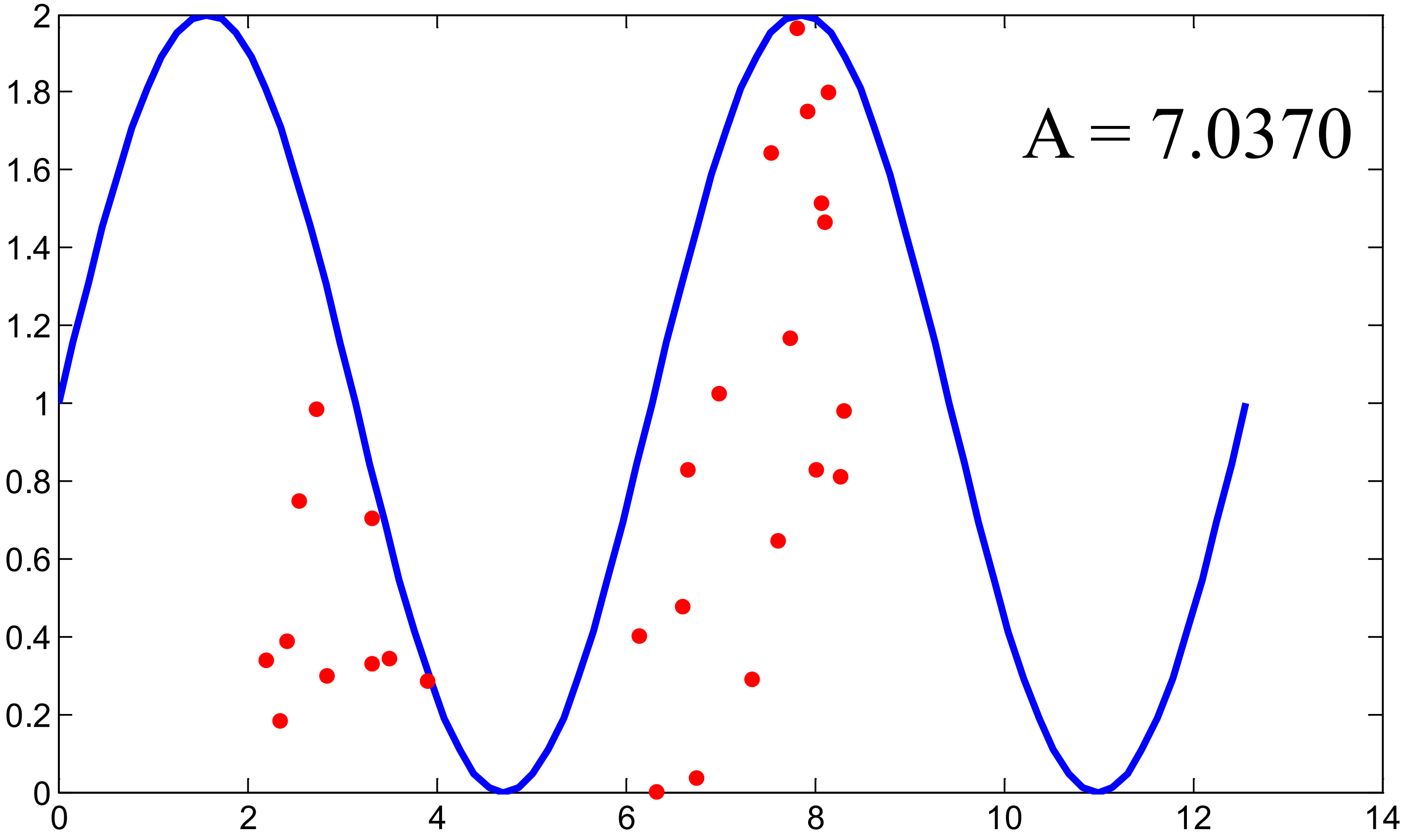
Plotting

```
t1 <- t[y < f_t]
y1 <- y[y < f_t]

t0 <- seq(0,10,0.1)
y0 <- f(t0)

ggplot() +
  geom_point(aes(x=t1,y=y1), size=0.5) +
  geom_path(aes(x=t0, y=y0), color='blue') +
  xlim(c(0,10))
```

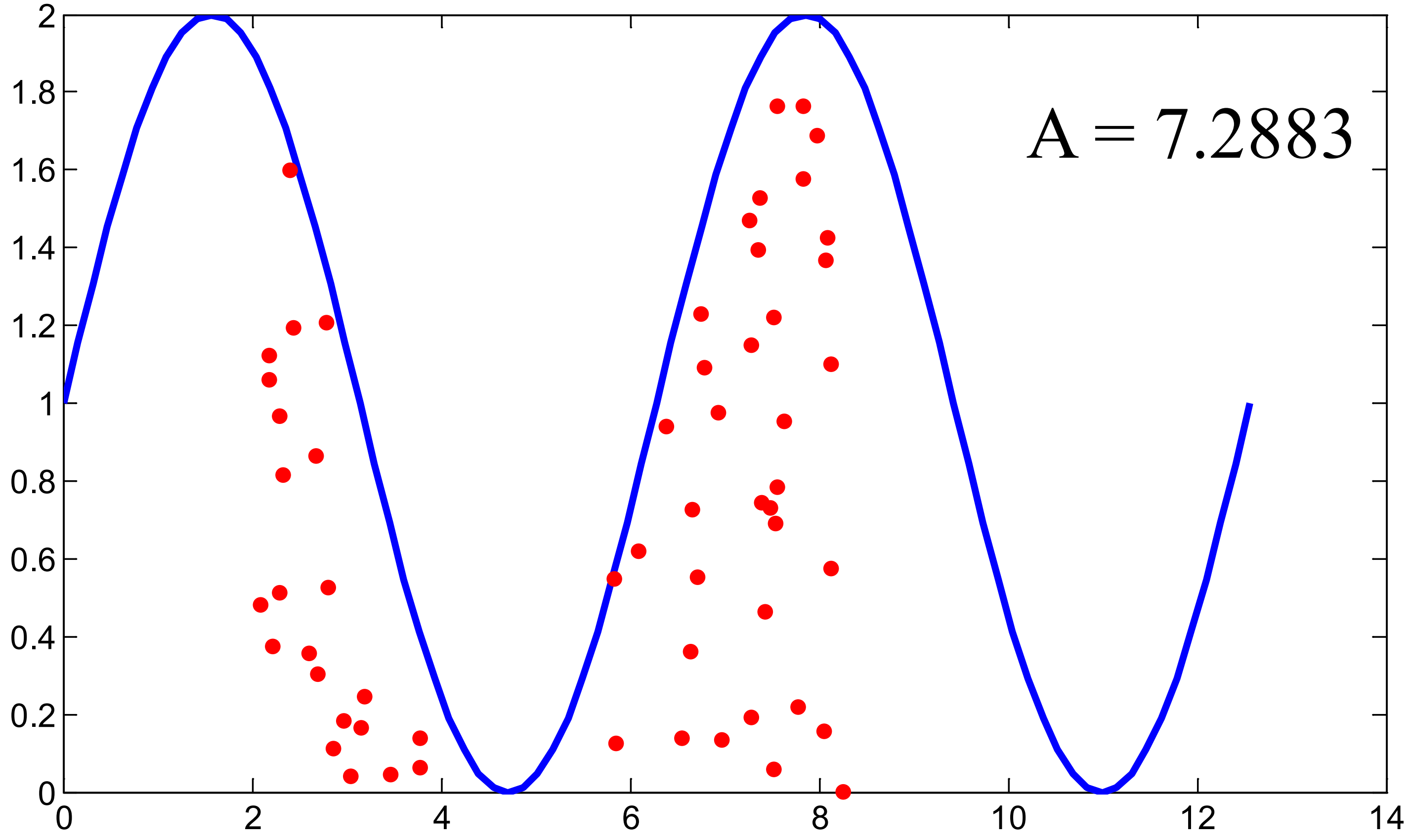
Monte Carlo simulation N=10



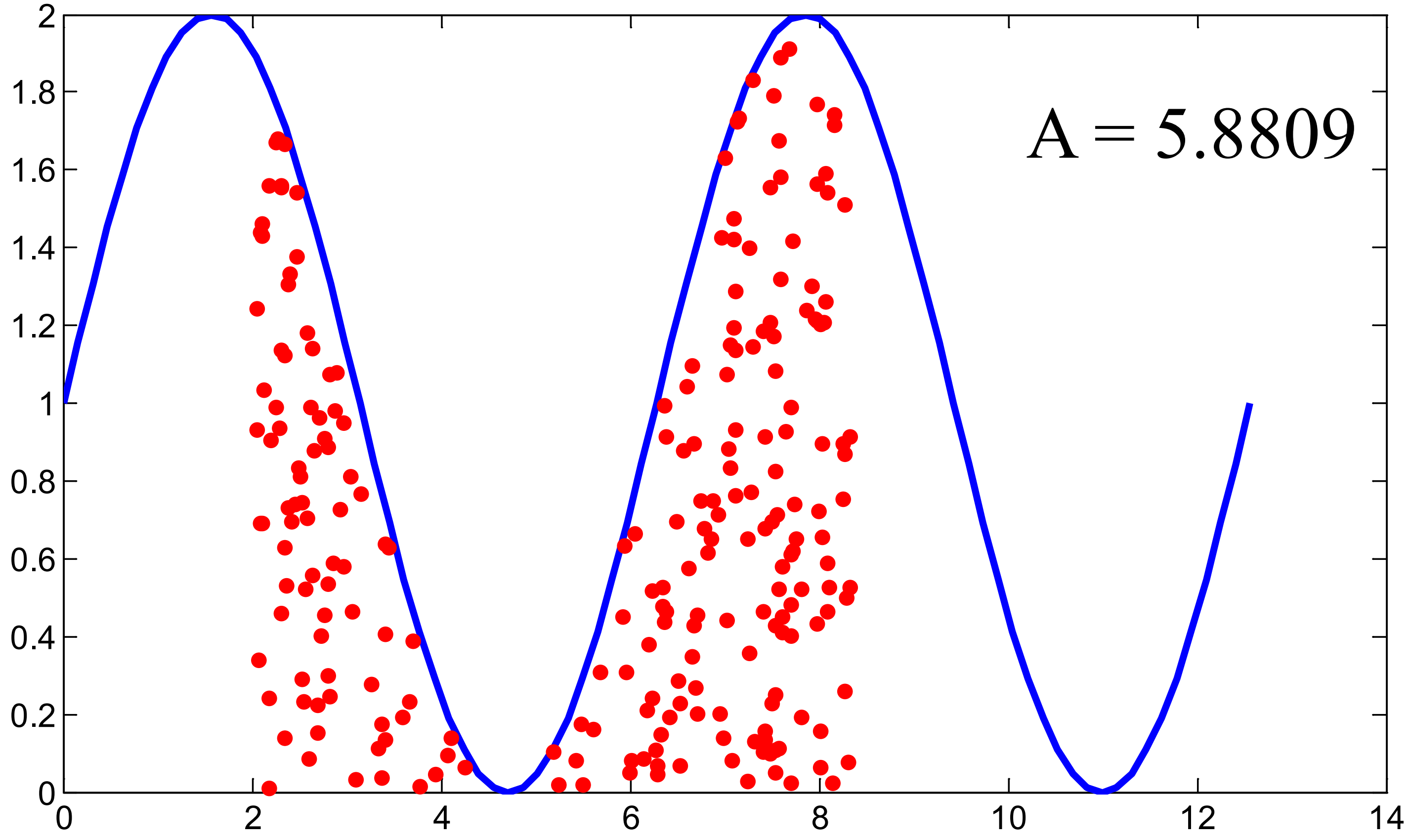
Monte Carlo simulation N=50



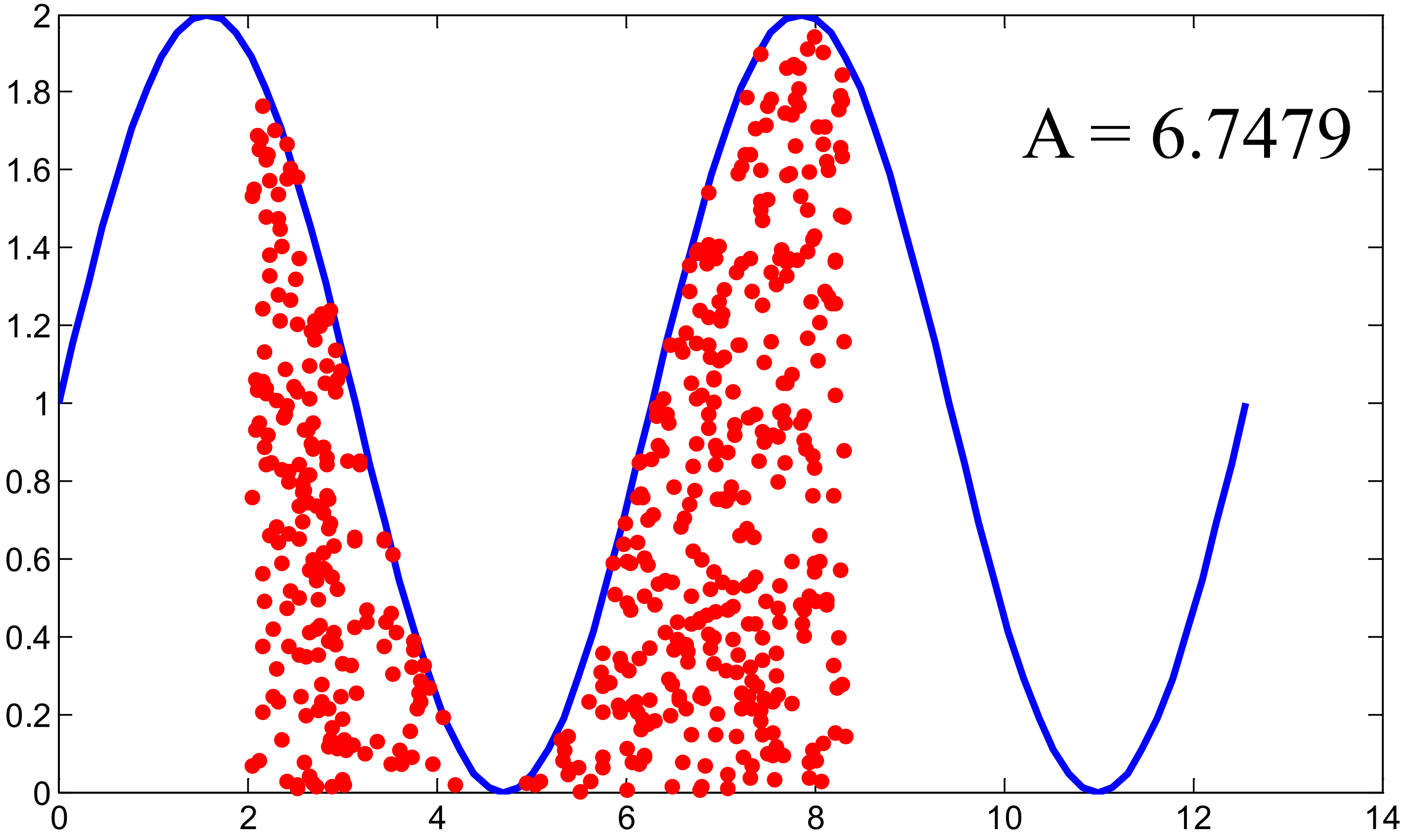
Monte Carlo simulation N=100



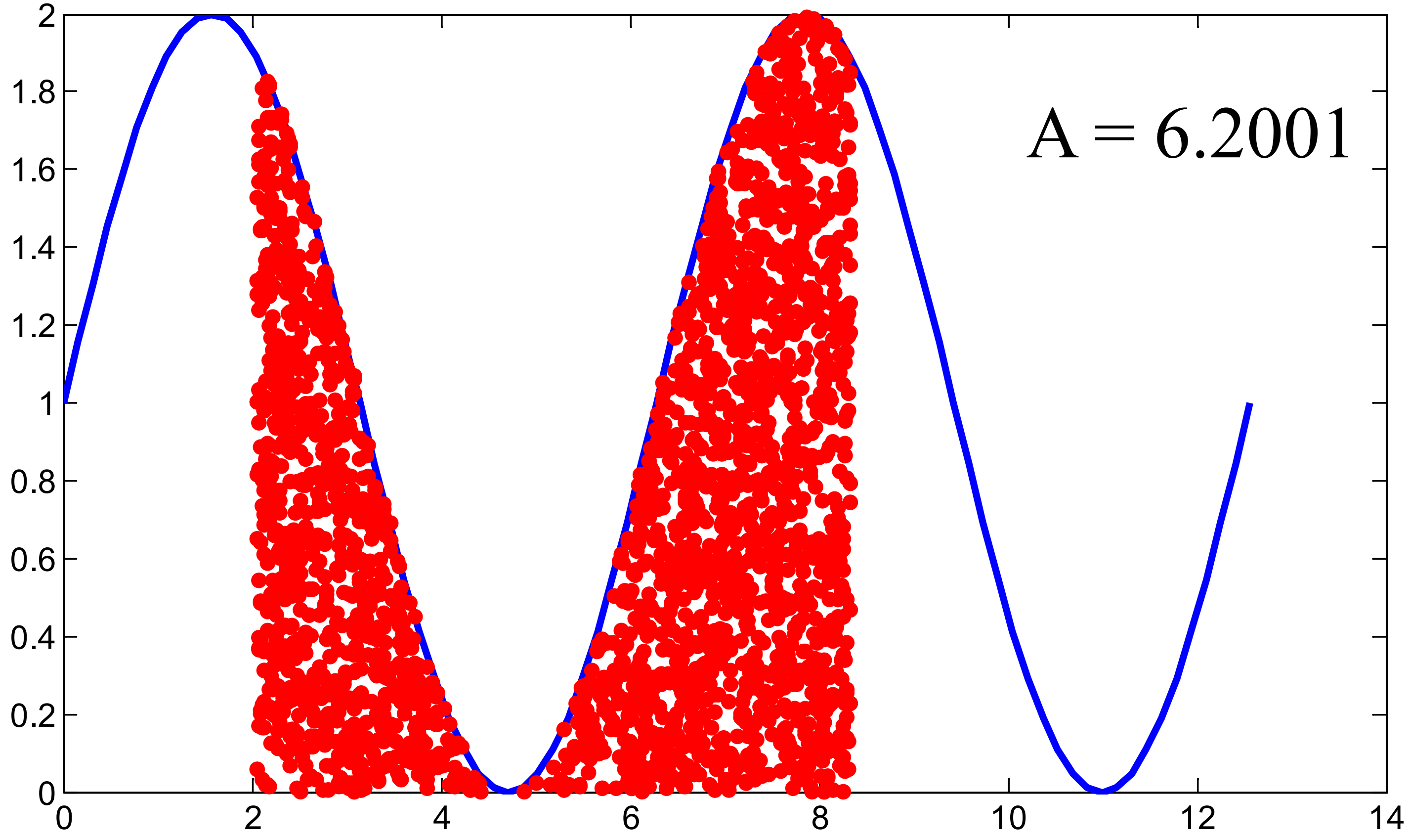
Monte Carlo simulation N=500



Monte Carlo simulation N=1000

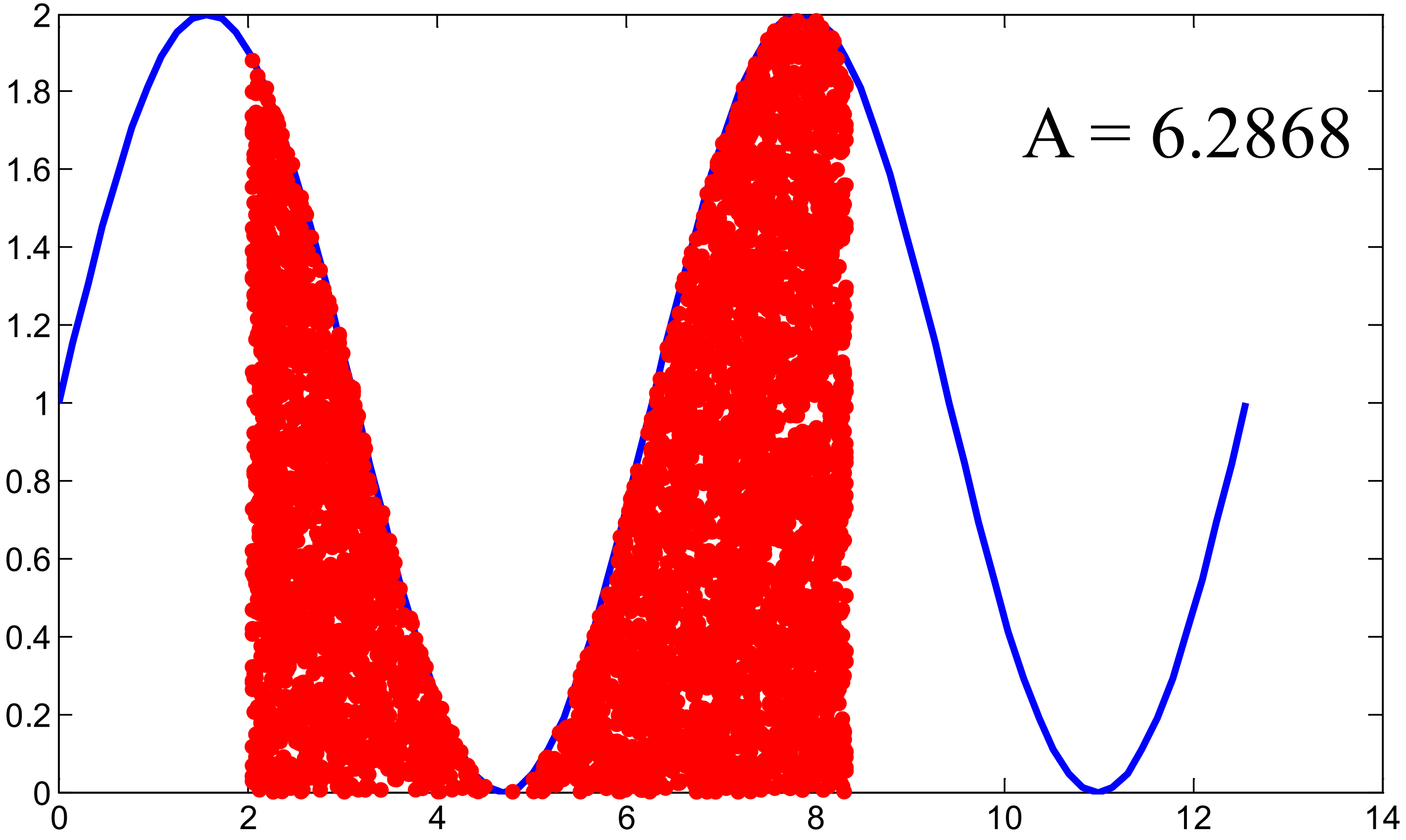


Monte Carlo simulation N=5000

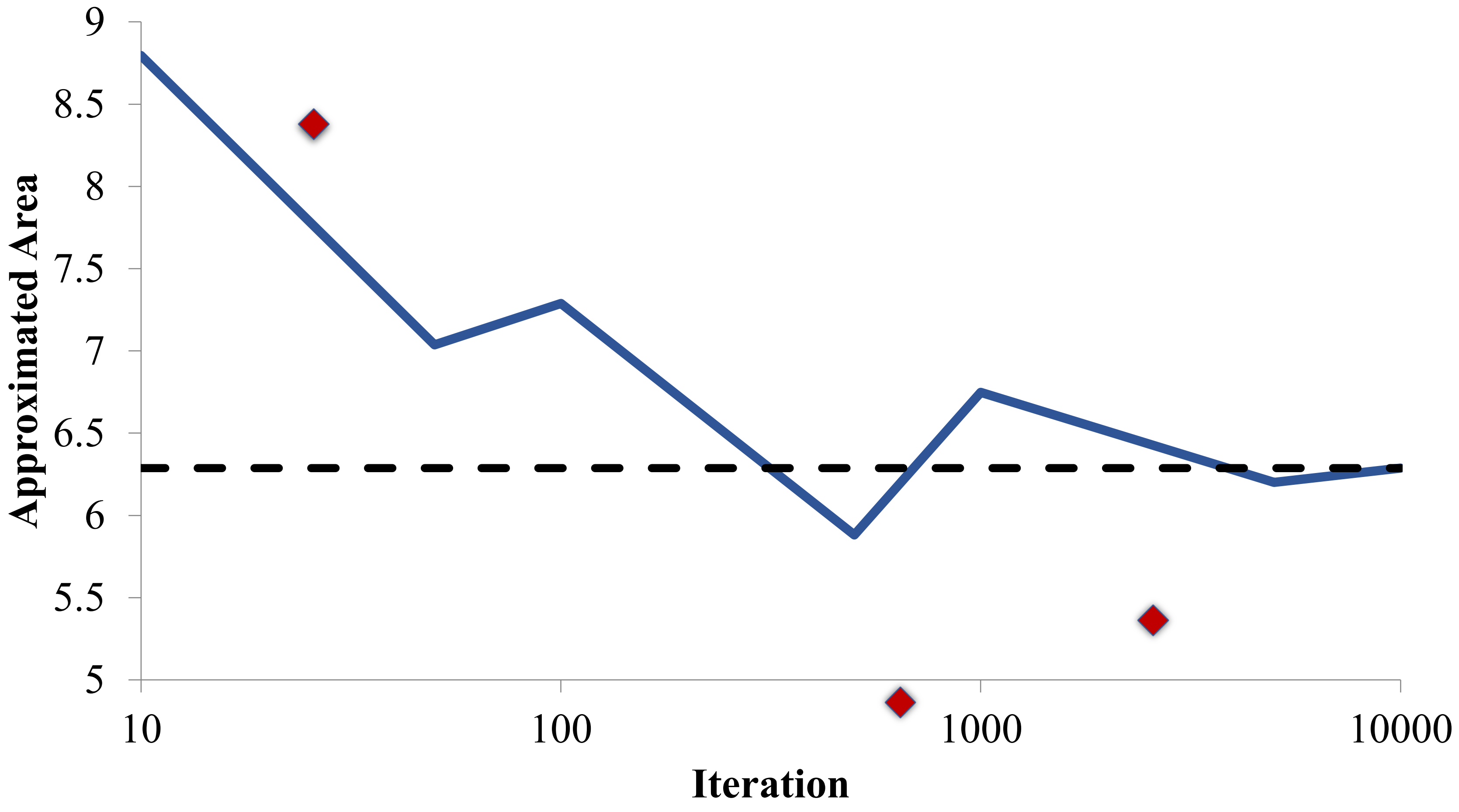


BIG DATA EXPERIENCE
KMUTT g-able

Monte Carlo simulation
N=10000



Integration approximation



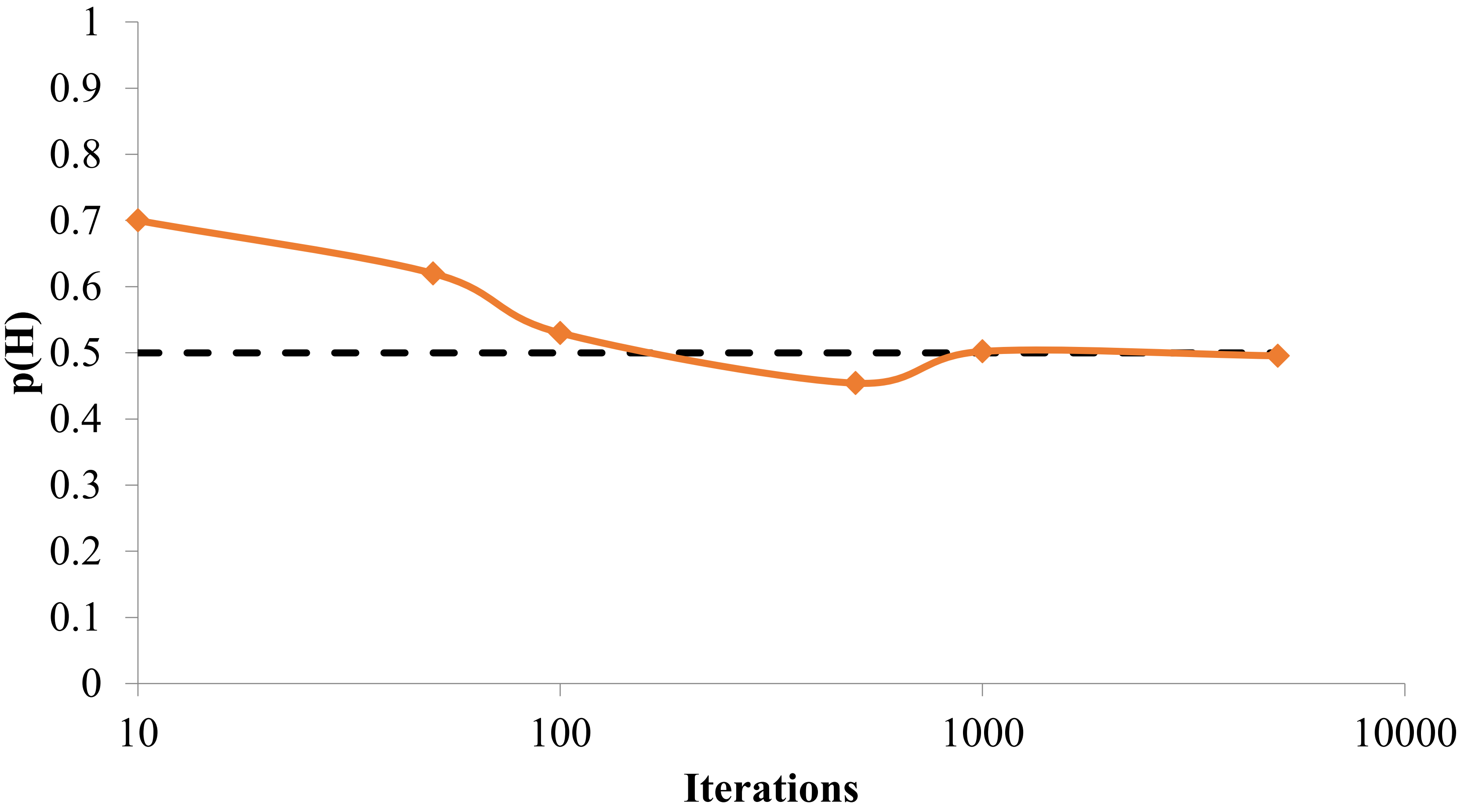
Simulating probabilistic behavior Example II: Coin tossing

For a fair coin, p(H) = p(T) = 0.5

Monte Carlo simulation of coin tossing

```
p <- runif(1000)
mean(p < 0.5)
```

Approximating probability



Important issues

- Replications in simulation

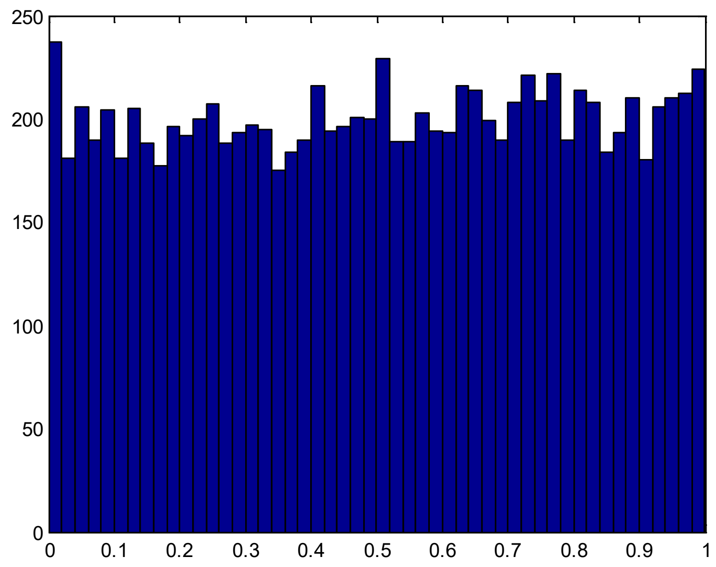
$y_1, y_2, \ldots, y_k$ k repetitions of N iterations

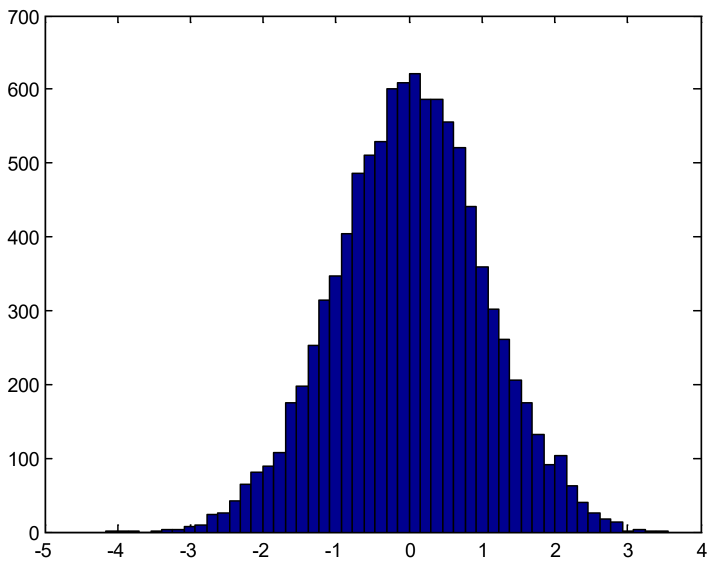
$y_N = (y_1+y_2+\ldots+y_k)/k$

- Random number generator

$$u \sim U[0,1]$$
$$u = F(x) \sim U[0,1]$$
$$x = F^{-1}(u)$$

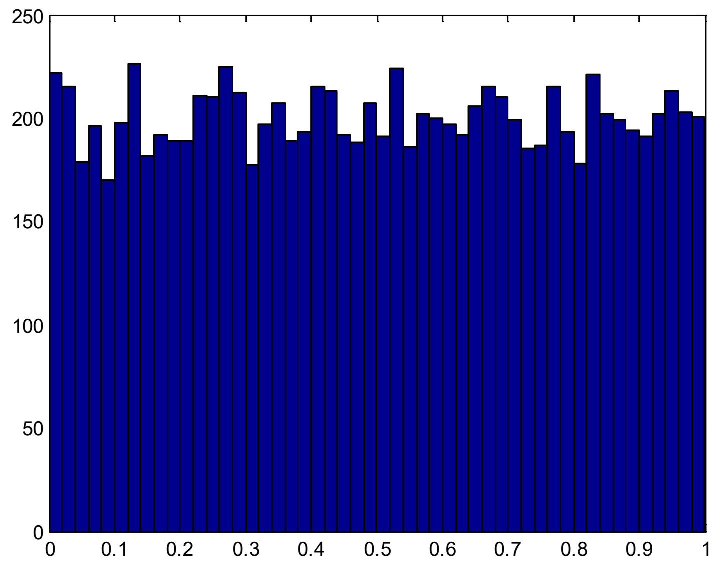
Generating random numbers of uniform distribution from U[0,1]

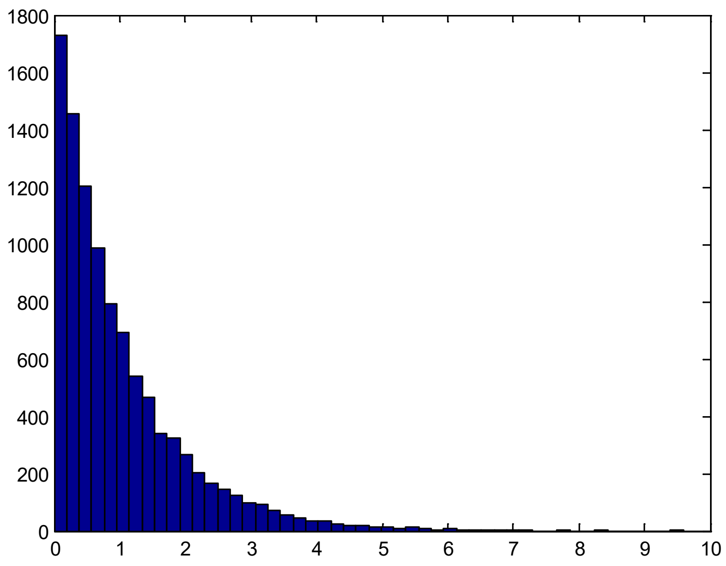




Generate uniform random numbers → Inverse normal dist. function → Normal distributed random numbers

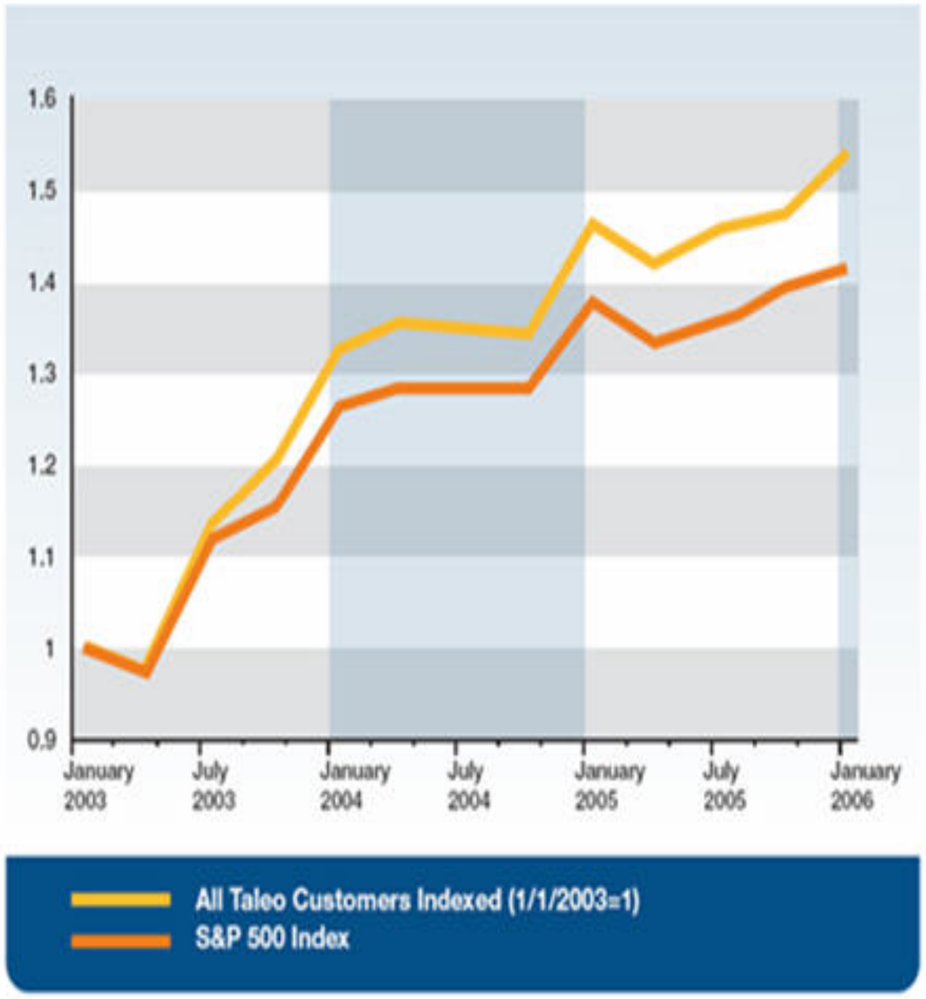
Generating random numbers of exponential distribution from U[0,1]





Generate uniform random numbers → Inverse exponential dist. function → Exponentially distributed random numbers

Example III: modeling stock market – simulating portfolio evaluation



Objectives

- To investigate various investment strategies
- To determine the strategy that likely to result in a good positive return over a long period of time
- The investor defines his/her decision method

Goals of this simulation

Portfolio

- Cash assets
- Stocks

Goals

- Each day, update the allocation of the asset between cash and stocks
- Increase the total value of the assets over time

Modeling investment strategy

- For our learning purpose, we assume a very simple investment strategy
- Only a single stock is considered

IF today stock price is higher

THEN spend 10% of cash asset to purchase shares of the stock

IF today stock price declines

THEN sell 10% of shares holding

Possible to include later

- Reversed strategy
- Waiting
- Threshold
- Brokerage commission
- Etc.

The stock market model Assumptions

- Today stock price is affected by change in the price of the stock of the previous day
- The change is a random number from a normal distribution with
  - σ = 1% of the previous day's price
  - μ = 10% of the previous day's price change

The stock market model Example

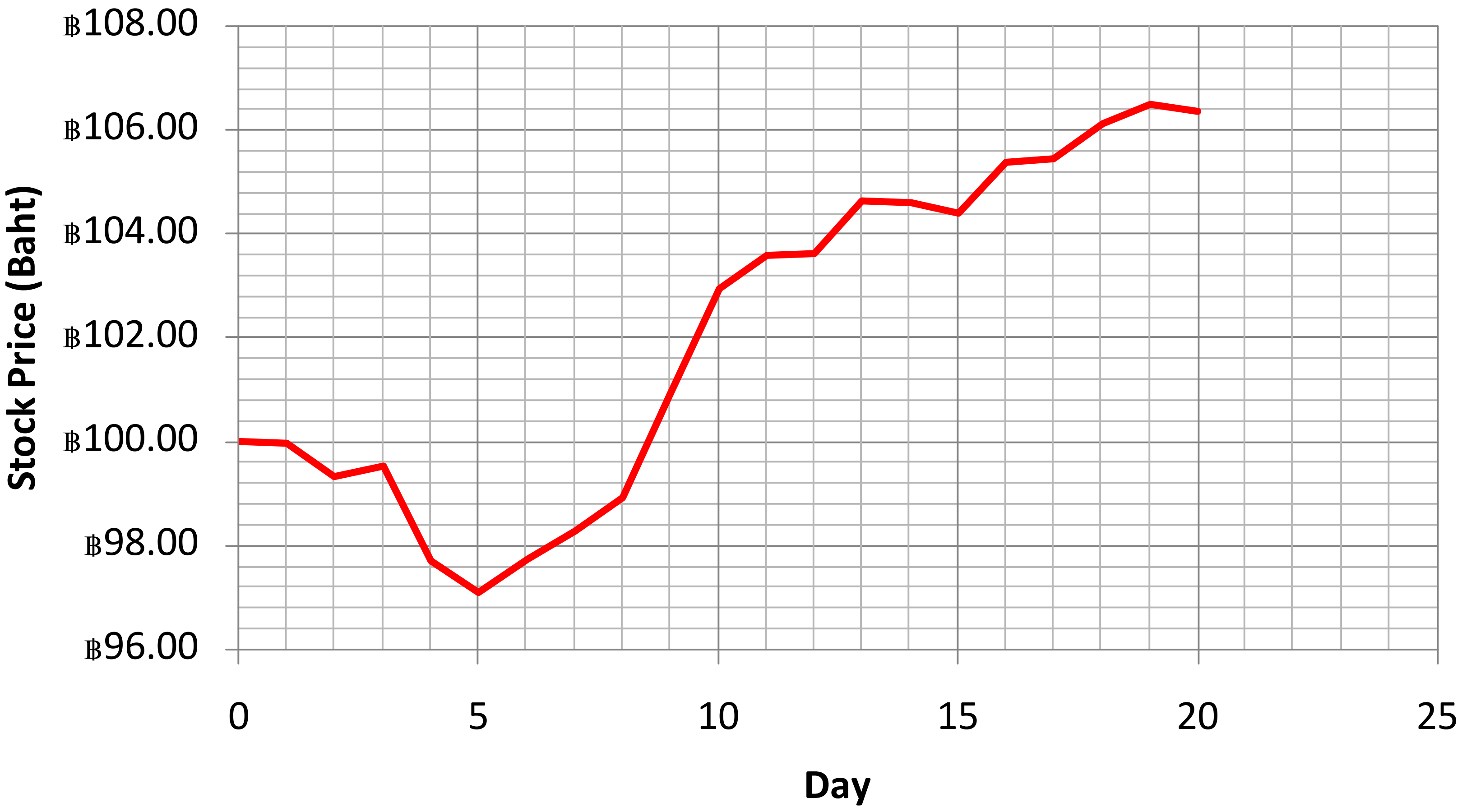
- Suppose that in previous day the stock went from 100 baht to 110 baht
- Today change would be sampled from a normal distribution with
  - a mean of 10%×10 baht = 1 baht
  - a standard deviation of 1%×100 baht = 1 baht

$$P_{n+1} = P_n + \varepsilon$$
$$\varepsilon \sim N(0.1(P_n - P_{n-1}), (0.01P_n)^2)$$

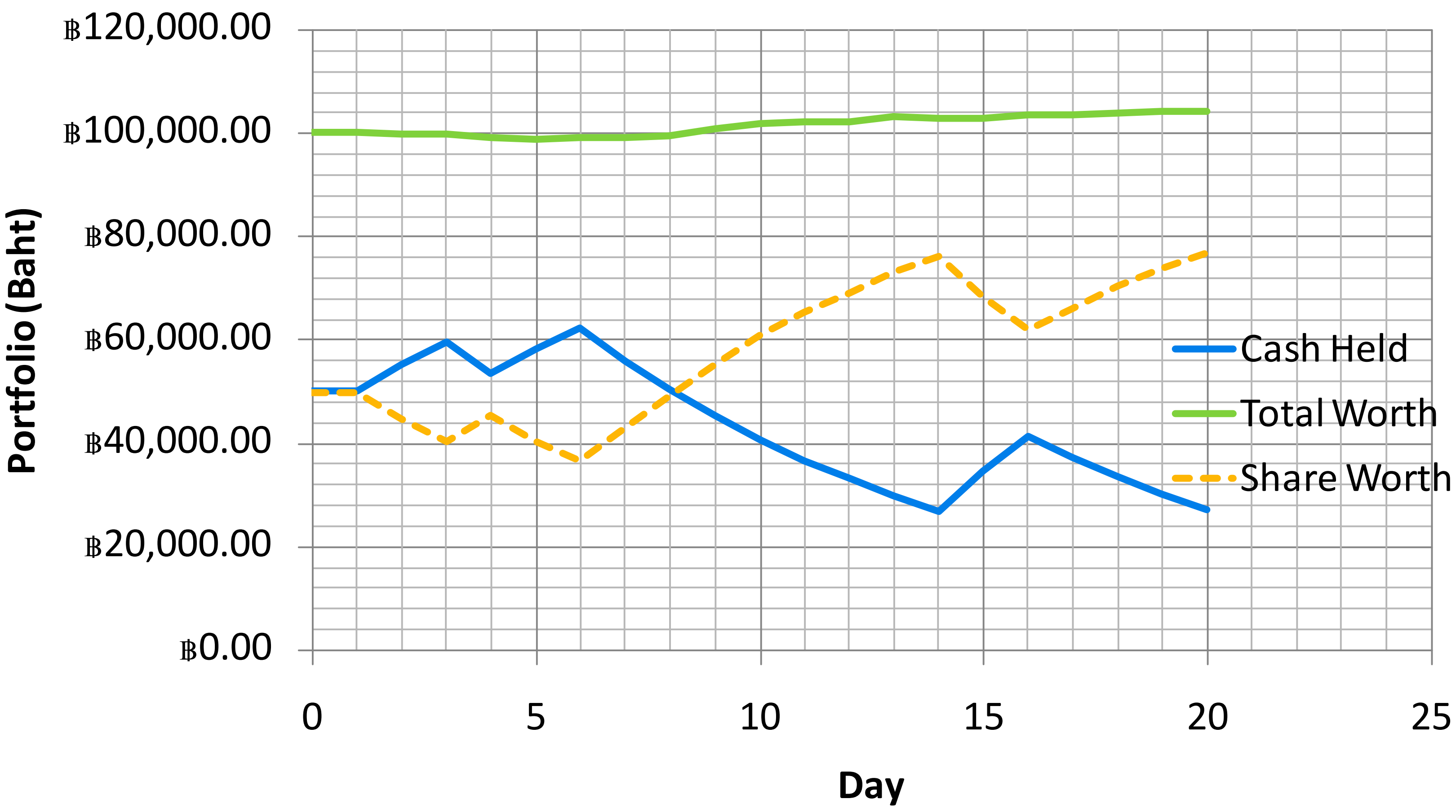
Simulating stock market model

Day	Stock Price	Share Held	Cash Held	Total Worth	Shares Purchase	Shares Sold	ΔPrice
0	฿100.00	500	฿50,000.00	฿100,000.00	0	0	
1	฿99.05	500	฿50,000.00	฿99,523.34	0	50	-฿0.95
2	฿99.31	450	฿54,952.33	฿99,643.82	55	0	฿0.27
3	฿100.29	505	฿49,490.04	฿100,133.97	49	0	฿0.97
4	฿99.68	554	฿44,576.08	฿99,796.66	0	55	-฿0.61
5	฿101.32	499	฿50,058.26	฿100,619.32	49	0	฿1.65
6	฿101.04	548	฿45,093.35	฿100,465.36	0	55	-฿0.28
7	฿100.40	493	฿50,650.76	฿100,145.68	0	49	-฿0.65
8	฿100.45	444	฿55,570.13	฿100,170.59	55	0	฿0.06
9	฿99.60	499	฿50,045.30	฿99,747.87	0	50	-฿0.85
10	฿99.46	449	฿55,025.52	฿99,682.99	0	45	-฿0.14

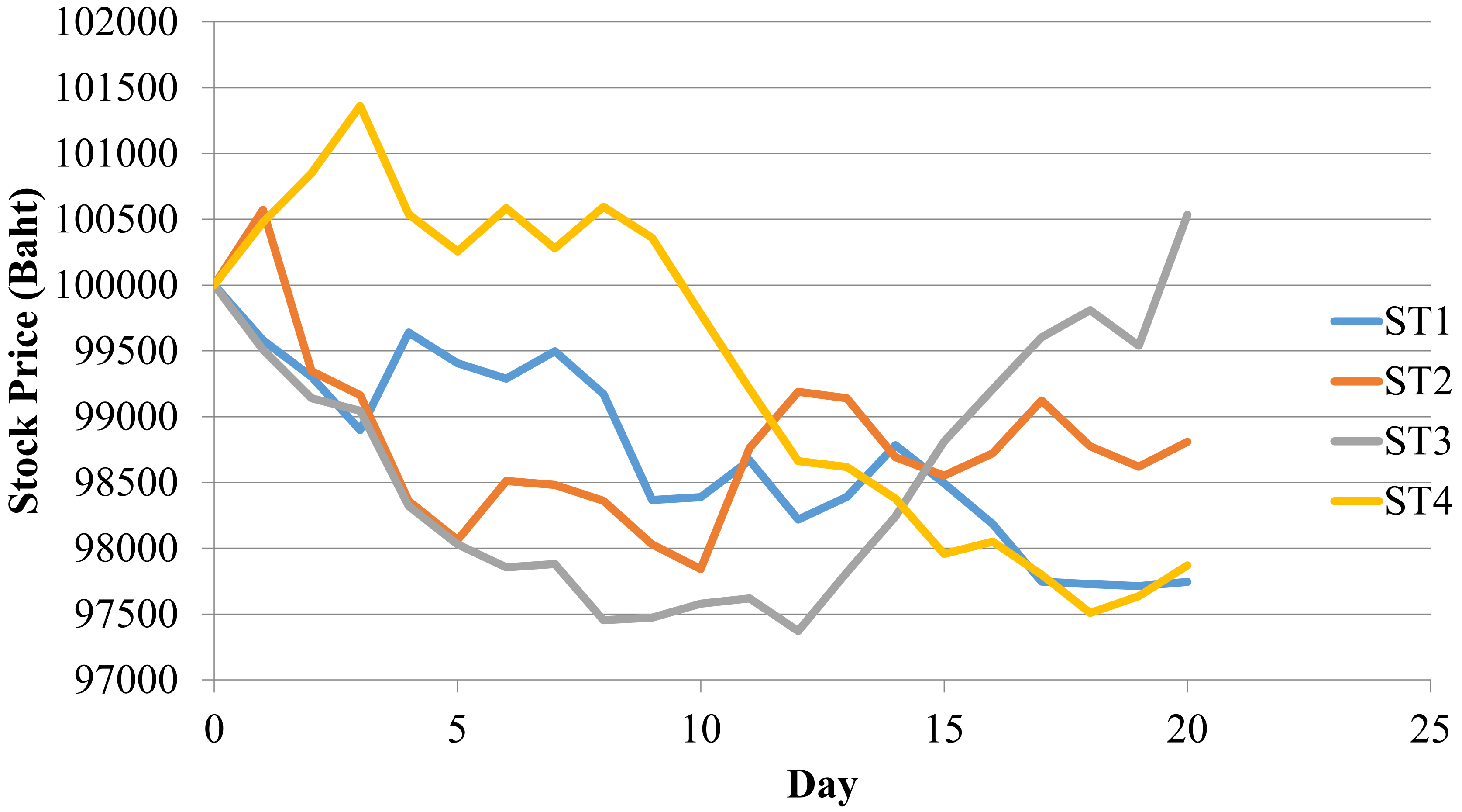
Simulated stock price



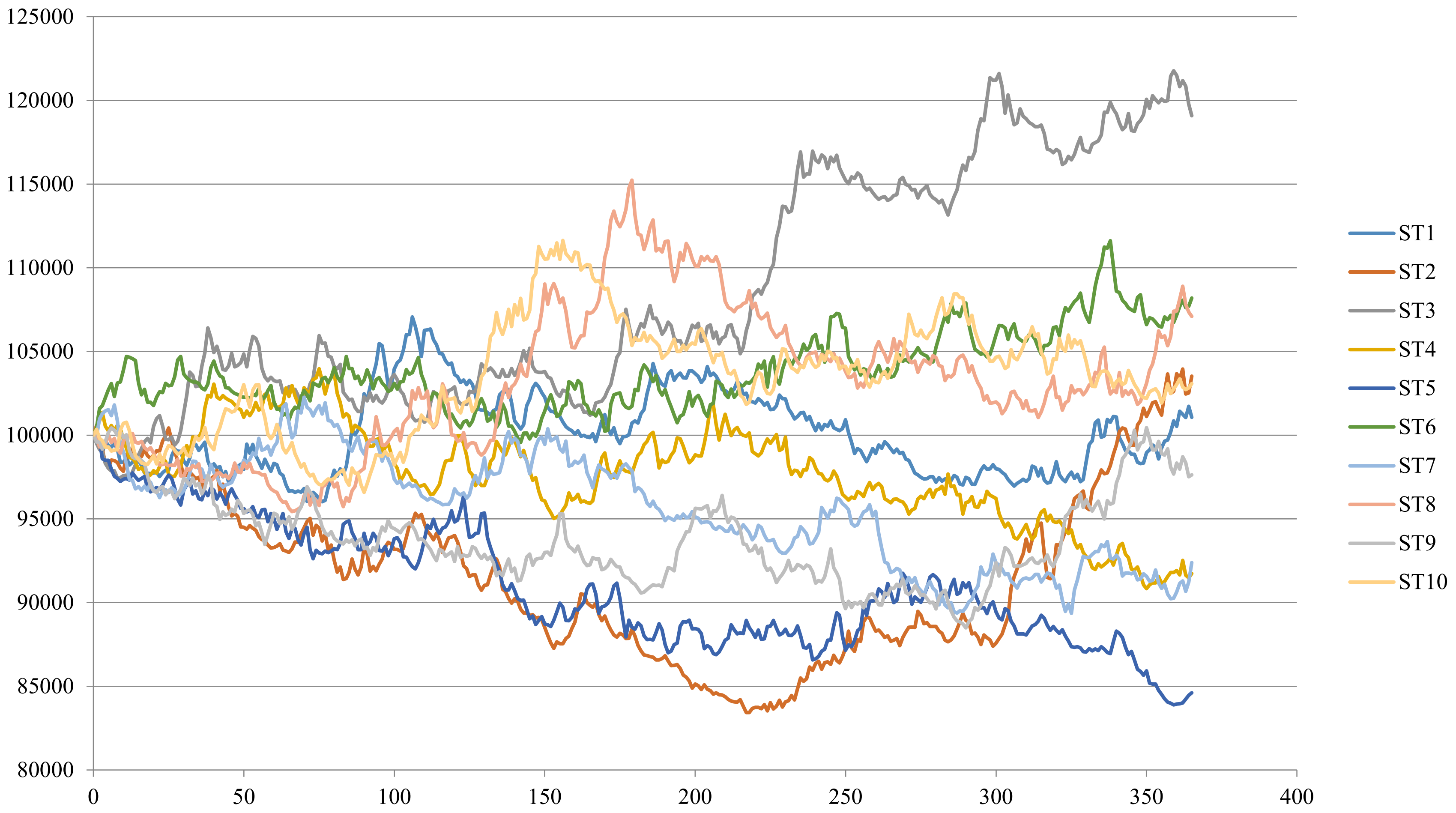
Simulated portfolio management



Randomness in Monte Carlo simulation – stock price



Randomness in Monte Carlo simulation – assets in portfolio



Example IV: portfolio evaluation

- Consider two stock, A and B
- Let $S_a(t)$ and $S_b(t)$ be the time t prices of A and B
- Let n_a and n_b be the owned units of A and B
- Let W(t) be the wealth at time t

Portfolio

- No trading between 0 to T

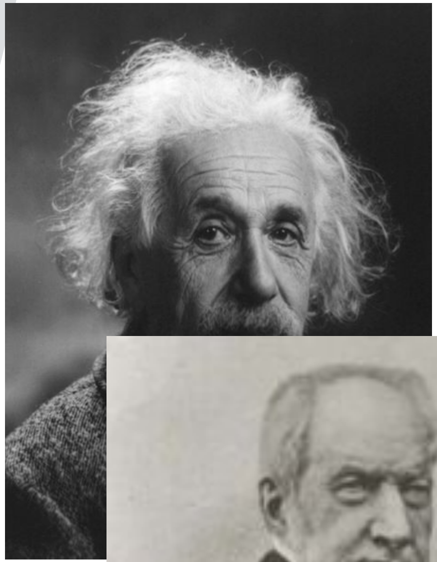
Initial

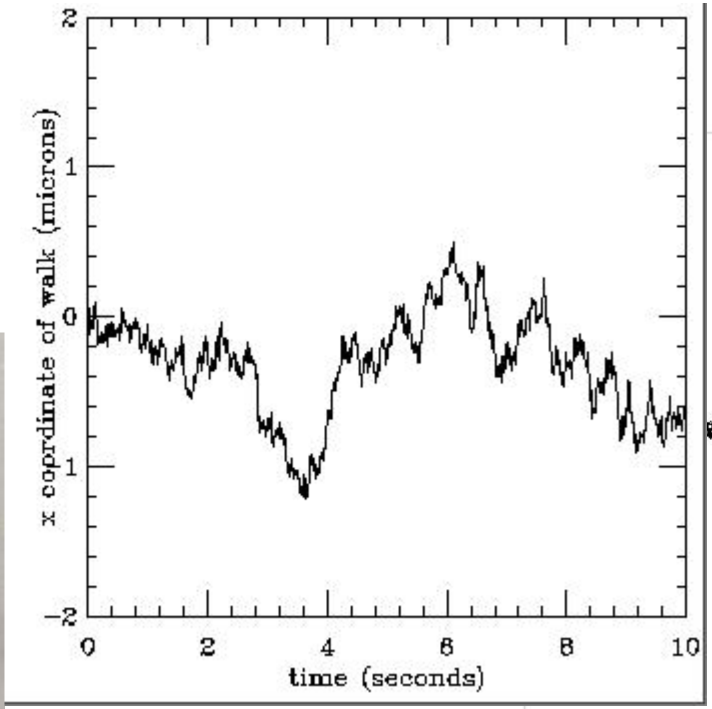
- At t=0, n_a units of A
  n_b units of B
  initial wealth $W(0) = n_a S_a(0)+n_b S_b(0)$

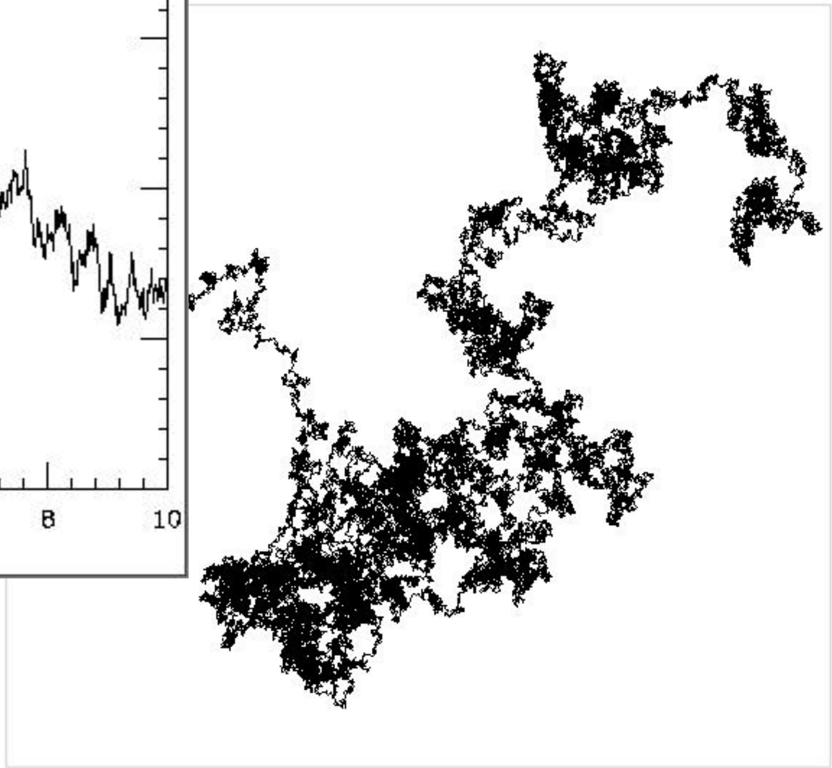
Terminal

- At t=T, n_a units of A
  n_b units of B
  initial wealth $W(T) = n_a S_a(T)+n_b S_b(T)$

Brownian motion







https://www.britannica.com/video/185377/Albert-Einstein-Description-motion-theory-size-Brownian

Brownian motion of stock prices

$$S_a(T) = S_a(0)\exp\left((\mu_a - \sigma_a^2/2)T + \sigma_a B_a(T)\right)$$
$$S_b(T) = S_b(0)\exp\left(\left(\mu_b - \sigma_b^2/2\right)T + \sigma_b B_b(T)\right)$$

where

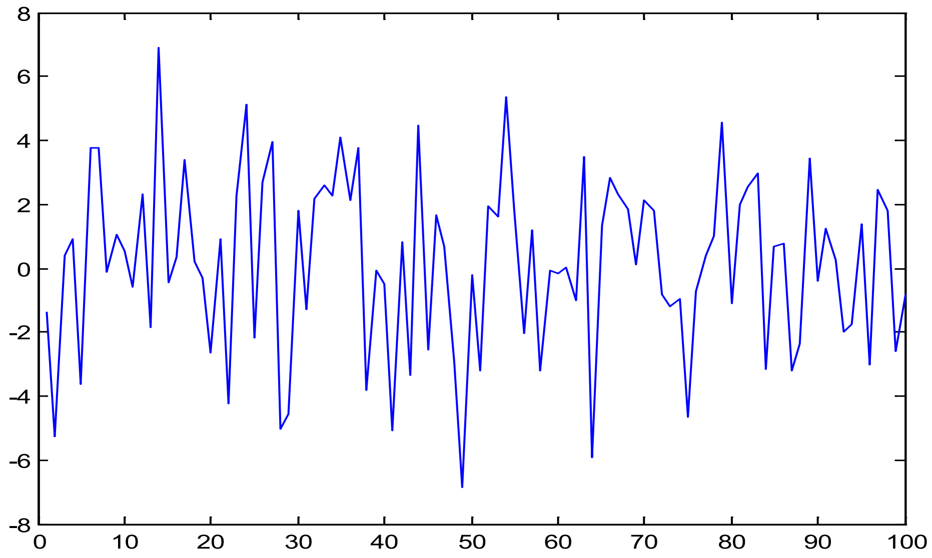
$B_a(t)$ and $B_b(t)$ are standard brownian motion

$B(t_i) = B(t_{i-1}) + X$

$X \sim N\,(0, \Delta t)$

Simulating standard Brownian motion

```
SET t0=0, B(t0)=0

FOR i=1 to n
  generate X ~ N(0,ti-ti-1)
  set B(ti) = B(ti-1)+X
END
```



Probability as an expected value of simulated trials

- Let L be the event that $W(T)/W(0) \le 0.9$

  $P(L) = E(I_L)$

  where

$$I_L(\mathbf{X}) = \begin{cases} 1 & \text{if } \frac{n_a S_a(T) + n_b S_b(T)}{n_a S_a(0) + n_b S_b(0)} \le 0.9 \\ 0 & \text{otherwise} \end{cases}$$

Set initial conditions

- Let assume these situations
  - $T = 0.5$ years
  - $\mu_a = 0.15$, $\mu_b = 0.12$, $\sigma_a = 0.2$, $\sigma_b = 0.18$
  - $S_a(0) = \$100$, $S_b(0) = \$75$
  - $n_a = n_b = 100$
- This implies $W(0) = \$17{,}500$

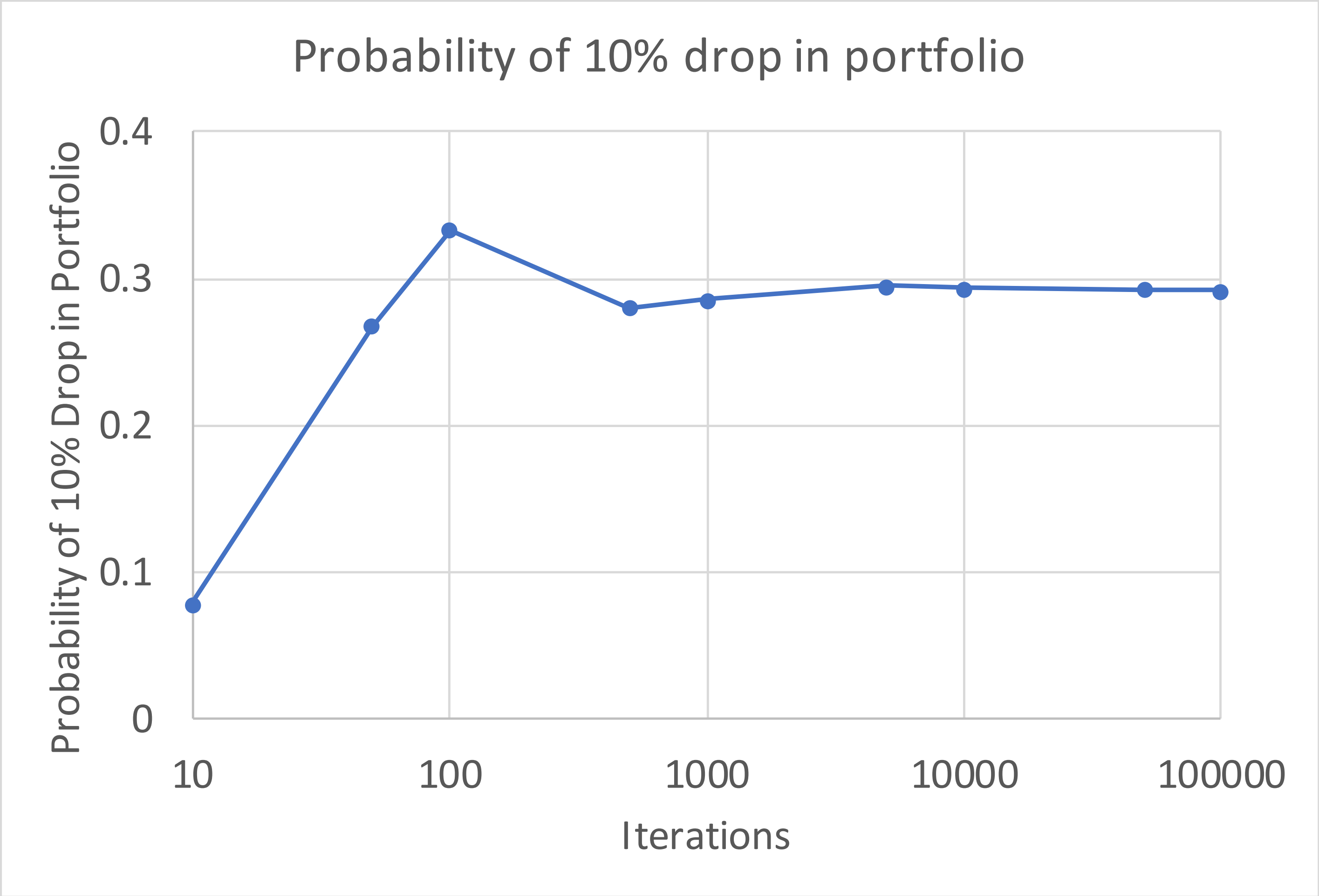
Monte Carlo simulation of $E(I_L)$

```
FOR i = 1 to N
 generate X^i = (S_a^i(T), S_b^i(T))
 compute IL(X^i)
END

E(IL) = (IL(X^1)+…IL(X^N))/N
```

Monte Carlo simulation of $E(I_L)$ in R

```
res <- NULL
N <- 1000 # number of simulation run
n <- 1000; tt <- 0.5; na <- 100; nb <- 100
S0a <- 100; S0b <- 75
mua <- .15; mub <- .12; siga <-.2; sigb <-.18
W0 <- na*S0a+nb*S0b
for (k in 1:N) {
  B1 <- cumsum(rnorm(n,0,1))   B1(t+1) = B1(t) + X(0,1)
  B2 <- cumsum(rnorm(n,0,1))
  STa <- S0a * exp((mua-(siga^2)/2)*tt + siga*B1)
  STb <- S0b * exp((mub-(sigb^2)/2)*tt + sigb*B2)
  WT <- na*STa + nb*STb
          as.numeric
  p <- mean(WT/W0 < 0.9)
  res <- c(res, p)
}
print(paste0("Prob of 10% drop is ",mean(res)))
```

Monte Carlo simulation result at least 10% drop in total asset



Thank you

Question?