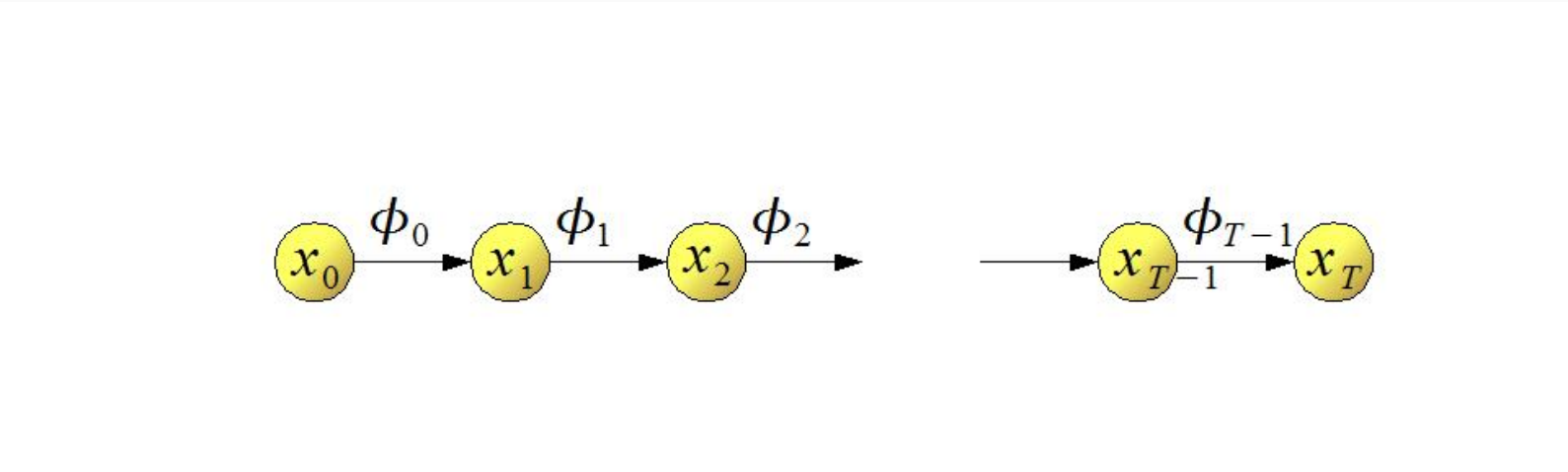
Dynamical system with randomness.

Provided a state x_t at the present time t, a system can often update it for the immediate future x_{t+1}. This suggests the evolution of states in the system

$$x_{t+1} = \phi_t(x_t)$$

where ϕ_t maps from all the possible states to themselves.



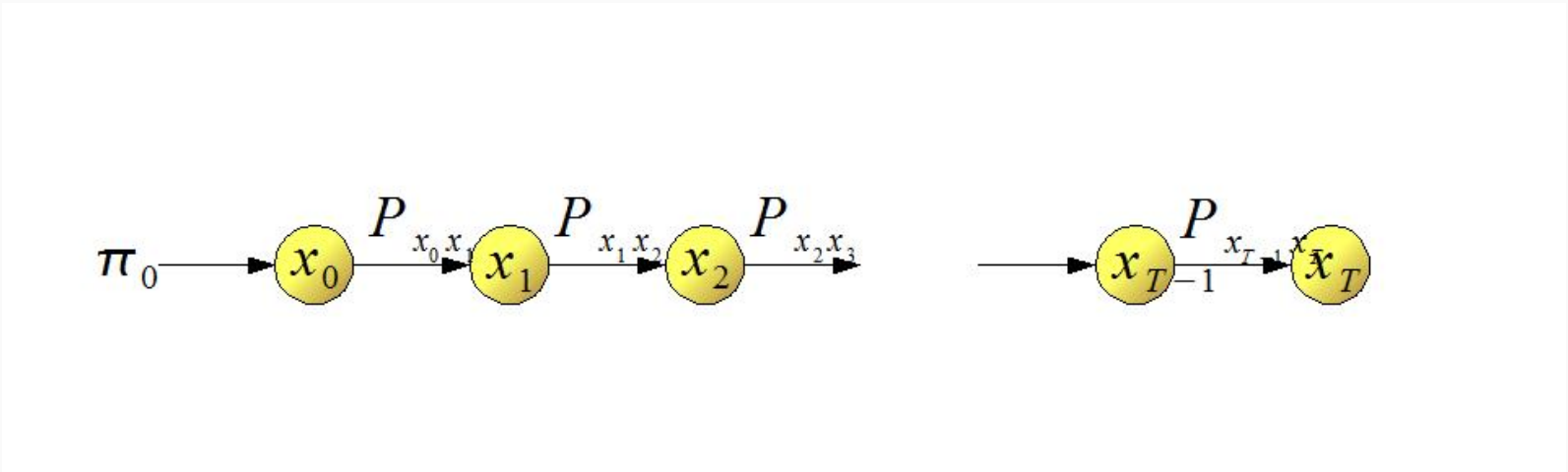
When the sequence $(\phi_t, t = 0, 1, \ldots)$ of maps is random, each state becomes a random variable X_t at $t = 0, 1, \ldots$, and $(X_t, t = 0, 1, \ldots)$ is collectively called a **stochastic process**.

Markov chain.

Furthermore, assuming that X_t's are discrete random variables, the stochastic process may be constructed by the **transition probability**

$$P_{xy} = \mathbb{P}(\phi_t(x) = y)$$

from $X_t = x$ to $X_{t+1} = y$ for finitely many states x and y, where $\mathbb{P}(A)$ denotes the probability of an event A. Together with initial probability mass function $\pi_0(x) = \mathbb{P}(X_0 = x)$, we can determine the distribution of a stochastic process $(X_t, t = 0, 1, \ldots)$, called **Markov chain**.



Markov property.

Let S be a discrete state space and let $(X_t,\ t = 0, 1, \ldots)$ be a stochastic process taking its value on S. A Markov chain can be characterized by the following property: It possesses the **Markov property** if the conditional probability of the n-step future X_{t+n} regardless of all the past X_s's for $s \leq t$

$$\mathbb{P}(X_{t+n} = x_{t+n} | X_s = x_s, s \leq t) = \mathbb{P}(X_{t+n} = x_{t+n} | X_t = x_t)$$

depends only upon the present state X_t. Moreover, if the above conditional probability does not depend on t then X_t is said to be **time-homogeneous**. Then one can construct a Markov chain $(X_t,\ t = 0, 1, \ldots)$ by the transition probability P_{xy} so that

$$P_{xy} = \mathbb{P}(X_{t+1} = y | X_t = x)$$

without referring to a random map ϕ_t.

Transition probability matrix.

When the system has a finite state space $S = \{1, \ldots, M\}$, the distribution of a Markov chain is determined by the **transition probability matrix**

$$P = \begin{bmatrix} P_{1,1} & P_{1,2} & \cdots & P_{1,M} \\ P_{2,1} & P_{2,2} & \cdots & P_{2,M} \\ \cdots & \cdots & \cdots & \cdots \\ P_{M,1} & P_{M,2} & \cdots & P_{M,M} \end{bmatrix}$$

Since P is a square matrix, we can also introduce the n-th power P^n whose (i,j)-entry is denoted by P_{ij}^n. It represents the n-**step transition probability**

$$P_{ij}^n = \mathbb{P}(X_n = j \mid X_0 = i)$$

which allows us to calculate the distribution π_n of X_n by

$$\pi_n(j) = \mathbb{P}(X_n = j) = \sum_{i=1}^{M} \pi_0(i) P_{ij}^n$$

R code: Manipulation of matrices.

Here matrices and their multiplication operation "%*%" are presented. By default data are viewed as a row vector. The function "`matrix()`" is used in order to convert it to a matrix. It returns the matrix with the specified number `ncol` of columns, starting from the first column to the last column.

```
P = matrix(c(0, 0, 0.6, 1, 0.1, 0.4, 0, 0.9, 0), ncol=3)
P
P %*% P
c(0.2, 0.8, 0) %*% P %*% P
```

The above computation should show the 2-step transition probability matrix P^2 and the probability distribution $\pi_2(j) = \mathbb{P}(X_2 = j)$, $j = 1, 2, 3$, given the row vector

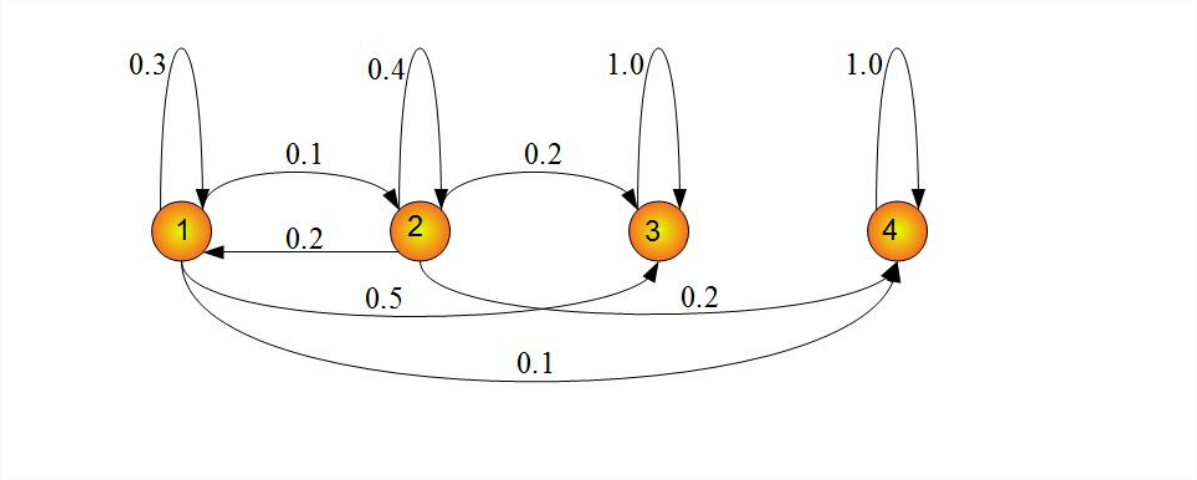
$$[\pi_0(1), \pi_0(2), \pi_0(3)] = [0.2, 0.8, 0]$$

of initial distribution.

Graph representation of transition probabilities.

A transition probability matrix P can be represented in terms of **directed graph** with positive weights on its edges. Here the state space S becomes the set of vertices of graph, and $E = \{(i,j) : P_{ij} > 0\}$ determines the collection of edges. The transition probability P_{ij} can be viewed as a positive weight assigned for each $(i,j) \in E$. For example,

$$P = \begin{bmatrix} 0.3 & 0.1 & 0.5 & 0.1 \\ 0.2 & 0.4 & 0.2 & 0.2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$ is equivalently formulated as follows.



Communication classes.

When we can find a series of edges

$$(i_0, i_1), (i_1, i_2), \ldots, (i_{n-1}, i_n)$$

we call it a **directed path** from i_0 to i_n. There is a directed path from i and j if and only if $P^n_{i,j} > 0$ for some n. We say that state j can be **reached from** state i, and write "$i \to j$" if $P^n_{ij} > 0$ for some $n \geq 0$, that is, if there is a directed path from i and j. The two states i and j are said to **communicate**, denoted by "$i \leftrightarrow j$," if we find $i \to j$ and $j \to i$. It is an equivalence relation, and therefore, the state space S can be partitioned into equivalent classes, which we call **communication classes**. In the previous example of graph representation, the communication classes are $\{1, 2\}$, $\{3\}$, and $\{4\}$.

Irreducibility.

The transition probability matrix P is said to be **irreducible** if it consists of only one communication class. A communication class C is called **recurrent** if no state outside can be reached from any state inside, that is,

$$P_{ij}^n = 0 \text{ for any } i \in C \text{ and } j \notin C, \text{ and for all } n \geq 0.$$

In particular, a state i is called **absorbing** if $P_{ii} = 1$ since the Markov chain has to stay at the state i forever (thus, "absorbed") once it reaches there. Otherwise, that is, if a communication class is **not** recurrent, we call it **transient**. In the previous example the class $\{1, 2\}$ is transient, and $\{3\}$ and $\{4\}$ are absorbing.

Periodicity of recurrent classes.

We say that a recurrent state i has the period d if $P_{ii}^n > 0$ only when $n = kd$ with some integer k. Furthermore, if state i has the period d and $i \leftrightarrow j$, then state j will have the same period d; thus, a recurrent class must have a common period d. If a recurrent class has the period $d \geq 2$ then it is called **periodic**; otherwise, (that is, if $d = 1$) it is said to be **aperiodic**. If a recurrent class C is aperiodic then

$$P_{ij}^n > 0 \text{ whenever } i, j \in C$$

for sufficiently large n. For example, $P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ is periodic, while $P = \begin{bmatrix} 0.1 & 0.9 \\ 1 & 0 \end{bmatrix}$ is aperiodic.

Eigenvalue and eigenvector.

A row vector $\mathbf{q}^T$ (which is a column vector $\mathbf{q}$ "transposed") is called a **left-eigenvector** if it satisfies $\mathbf{q}^T P = \lambda \mathbf{q}^T$. Then λ is called the corresponding **eigenvalue**. The eigenvalue $\lambda = 1$ is known to be the largest in absolute value for any transition probability matrix P. Furthermore, Frobenius showed that the corresponding left-eigenvector $\mathbf{q}^T = [q_1 \cdots q_M]$ is strictly positive (i.e., $q_i > 0$ for all $i = 1, \ldots, M$), and can be chosen as the unique solution to the equations

$$q_j = \sum_{i=1}^{M} q_i P_{ij} \text{ for all } j = 1, \ldots, M \text{ and } \sum_{i=1}^{M} q_i = 1.$$

Thus, q_i is viewed as probability mass function $\pi_\infty(i)$, and called the **stationary distribution**.

Perron-Frobenius theorem.

Historically the discovery of fundamental theorem was made by Perron in 1907. Suppose that a transition probability matrix P is aperiodic and irreducible. Then P has a unique left-eigenvector $\mathbf{q}^T = [q_1 \cdots q_M]$ corresponding to the eigenvalue $\lambda = 1$. It represents the stationary distribution $\pi_\infty(i) = q_i$ and

$$\lim_{n\to\infty} P_{ij}^n = \pi_\infty(j) \quad \text{for all } i = 1, \ldots, M.$$

Moreover, if λ_2 is the second largest eigenvalue of P in absolute value, for any $|\lambda_2| < \rho < 1$ we can find $C > 0$ such that

$$\max_{1\le i,j\le M} |P_{ij}^n - \pi_\infty(j)| < C\rho^n \quad \text{for all } n = 1, 2, \ldots.$$

R code: Eigenvector and stationary distribution.

Compare the probability distribution $\pi_n = \pi_0 P^N$ at the N-th step with the normalized eigenvector $\mathbf{u}$ corresponding to $\lambda = 1$. Note that `eigen()` calculates the left-eigenvectors when the transposed matrix `t(P)` is applied.

```
N = 20
A = P
for(i in 1:N) A = A %*% P
A
c(0.2, 0.8, 0) %*% A
uu = as.numeric(eigen(t(P))$vectors[,1])
uu / sum(uu)
```

You may change the initial distribution π_0 and the size N, and observe consistently a similar result.

Markov chain convergence theorem.

We call a Markov chain $(X_t, t = 0, 1, \ldots)$ (and its transition probability P) **ergodic** if P is irreducible and aperiodic. Now suppose that we have devised an ergodic Markov chain whose stationary distribution is π. Then the following convergence theorem is applicable: The stationary distribution π is the unique distribution so that

$$P_{ij}^n = \mathbb{P}(X_n = j | X_0 = i) \to \pi(j) \qquad \text{as } n \to \infty,$$

regardless of the choice for an initial state i. In terms of sample path we obtain

$$X_t \xrightarrow{\mathcal{L}} \pi \quad \text{as} \quad t \longrightarrow \infty.$$

where "$\mathcal{L}$" indicates the convergence in the "law" of probability distribution.

Emergence of Markov chain Monte Carlo.

In practice the "state" space S for $\pi(x)$ is very large or even uncountable. It is often a subset of $\mathbb{R}^n$ (e.g., a parameter space for posterior distribution π in Bayesian statistics), or it has a complex discrete structure (e.g., an Ising model in statistical mechanics). For such models neither sampling via inverse probability transform (applicable only for one-dimensional space $\mathbb{R}$) nor resampling by rejection methods [unable to find an upper bound $\pi(x) \leq c\rho(x)$] are practical. Instead, by way of Markov chain convergence theorem one may obtain a sample from π by observing a Markov chain X whose stationary distribution is π.

$$X_t \xrightarrow{\mathcal{L}} \pi \quad \text{as} \quad t \longrightarrow \infty.$$

This methodology gives a way to break the limitation of Monte Carlo simulation. But how can we devise such a Markov chain?

Metropolis-Hastings algorithm (MHA).

Let π be a probability mass function (pmf) of interest on a state space S, and let $Q(x, y)$ be a transition probability from x to y. Define an acceptance probability by

$$A(x, y) := \min \left\{ \frac{\pi(y)Q(y, x)}{\pi(x)Q(x, y)}, 1 \right\}$$

for a move from x to y satisfying $Q(x, y) \neq 0$ and $Q(y, x) \neq 0$.

Metropolis-Hastings Algorithm:

1. Choose an initial state X_0 at time $t = 0$.
2. Suppose that $X_t = x$ at time t. Then pick y according to $Q(x, \cdot)$.
3. Move to $X_{t+1} = y$ with the probability $A(x, y)$, or stay $X_{t+1} = x$ with the probability $(1 - A(x, y))$.

In calculation of acceptance probability $A(x, y)$ a symmetric Q can be chosen [i.e., $Q(x, y) = Q(y, x)$]. In practice $A(x, y) = \pi(y)/\pi(x)$ may be easily obtained.

Transition probability for MHA.

One should choose the transition probability Q to be ergodic on S, and call it a **proposal** chain. In terms of a stochastic process $(X_t, t = 0, 1, \ldots)$ the Metropolis-Hastings algorithm generates a Markov chain, and it has a transition probability from x to y by

$$P(x,y) = \begin{cases} Q(x,y)A(x,y) & \text{if } x \neq y; \\ Q(x,x) + \sum_{z \in S} Q(x,z)(1 - A(x,z)) & \text{if } x = y. \end{cases}$$

Then P becomes ergodic and "time-reversible" transition probability with respect to π. Thus, P satisfies

$$\pi(x)P(x,y) = \pi(y)P(y,x)$$

and it has the stationary distribution π.

What is time-reversibility?

Let π be a probability distribution on S. We call π **stationary** if

$$\sum_{x \in S} \pi(x) P(x, y) = \pi(y)$$

for all $y \in S$. A Markov chain with transition probability $\tilde{P}$ is said to be **time-reversed** if the **detailed balance**

$$\pi(x) P(x, y) = \pi(y) \tilde{P}(y, x)$$

holds between P and $\tilde{P}$. Conversely if the probability distribution π satisfies the detailed balance between P and $\tilde{P}$, then π becomes stationary for P since

$$\sum_{x \in S} \pi(x) P(x, y) = \pi(y) \sum_{x \in S} \tilde{P}(y, x) = \pi(y)$$

Moreover, P is called a *reversible* when $P = \tilde{P}$.

Time-reversibility for MHA.

The detailed balance clearly holds for either $x = y$ or $\pi(x)Q(x,y) = \pi(y)Q(y,x)$. Suppose that $x \neq y$, and that $\pi(x)Q(x,y) > \pi(y)Q(y,x)$. Then we can observe

$$A(x,y) = \frac{\pi(y)Q(y,x)}{\pi(x)Q(x,y)} \text{ and } A(y,x) = 1,$$

and therefore, we obtain

$$\begin{aligned}\pi(x)P(x,y) &= \pi(x)Q(x,y)A(x,y) \\ &= \pi(y)Q(y,x) = \pi(y)P(y,x).\end{aligned}$$

It also holds for the case that $\pi(x)Q(x,y) < \pi(y)Q(y,x)$ by symmetry.

Random beta-walk.

Here we will introduce a random walk on $(0, 1)$ in which the next step X_{t+1} is determined by the beta distribution with parameter $\alpha_1 = \delta + \theta X_t$ and $\alpha_2 = \delta + \theta(1 - X_t)$. These parameters can control the behavior of walk.

- A smaller δ keeps a sample path closer to either of the boundary points, 0 or 1.
- The larger the value θ is, the smaller a move by each step becomes.

Thus, $0 < \delta < 1$ will change the shape of stationary distribution of the random walk, and $\theta > 0$ will influence the speed of convergence of random walk. We will use this random walk as a proposal Markov chain on $(0, 1)$.

R code: Sample paths by beta-walk.

Install the package "coda" and download the R code "bwalk.r". It generates sample paths of the beta random walk by choosing sample.size of multiple runs (and it sets sample.size=4 by default).

- Choose a different choice of δ and θ.
- Change the running time and see how long it takes to display a stationary behavior.

traceplot() shows multiple sample paths in different colors.

```
library(coda)
source("bwalk.r")
par(mfrow=c(2,1))
bw1.sample <- rwalk(move=bmove, run.time=400, delta=0.8, theta=20)
traceplot(bw1.sample)
bw2.sample <- rwalk(move=bmove, run.time=400, delta=0.1, theta=20)
traceplot(bw2.sample)
```

R code: A long run behavior.

A long run behavior can be observed from the distribution of X_t's with times t toward the end of runs.

- Change `running.time` and choose `start` and `end` in `window()` toward the end of runs. See if the distribution of X_t's is different.
- Obtain the distribution of X_t's for a different choice of δ and θ.
- Change the initial point by adding "`init.state=1`", and see whether it affects the long run behavior.

`densplot()` displays a plot of the density estimate from multiple sample paths. `window()` determines where we pick sample paths for X_t's.

```
bw1.sample <- rwalk(move=bmove, run.time=400, delta=0.8, theta=20)
densplot(window(bw1.sample,start=301,end=400))
bw2.sample <- rwalk(move=bmove, run.time=400, delta=0.1, theta=20)
densplot(window(bw2.sample,start=301,end=400))
```

R code: MHA by beta-walk.

We can use a random beta-walk as a proposal chain, and run Metropolis-Hastings Algorithm (MHA) whose target distribution (`ff=mn`) is a normal mixture density truncated on $[0, 1]$.

```
source("nm.r")
source("metro.r")
mh.sample <- bwalk.metro(ff=nm, run.time=400, delta=0.8, theta=20)
traceplot(mh.sample)
densplot(window(mh.sample,start=301,end=400))
x = seq(0,1,by=0.01)
lines(x, nm(x), lty=2, col='red')
```

Change the running time and the parameters for random beta walk, and see how the distribution of sample paths by MHA differs from the target distribution of normal mixture (indicated by a red line).

Activities and report questions.

1. Explain ergodicity in the context of Markov chains on a finite state. Discuss the importance of Perron-Frobenius theorem in MCMC.
2. Explain how MHA works. Use the R code presented in the class, and demonstrate that MHA produces the target distribution of interest instead of the original distribution obtained from random beta-walk.
3. Run the following code and discuss the choice of initial state and running time in MCMC methodology.

```
mh1.sample <- bwalk.metro(ff=nm, init.state=0, run.time=50, delta=0.8,
theta=20, sample.size=50)
densplot(window(mh1.sample,start=31,end=50))
mh2.sample <- bwalk.metro(ff=nm, init.state=1, run.time=50, delta=0.8,
theta=20, sample.size=50)
densplot(window(mh2.sample,start=31,end=50))
```

MCMC in practice.

A design of MCMC in practice should include a convergence analysis by simulating multiple runs. Furthermore, in order to diminish the effect of initial states it is common to discard the first half $X_i^{(j)}, i = 1, \ldots, n_1$, of each sequence $j = 1, \ldots, n_2$ by seting `windows()` for the second half $X_i^{(j)}, i = n_1 + 1, \ldots, 2n_1$. The practice of discarding early iterations in MCMC is referred to as burn-in. The convergence may be inferred by comparing the multiple runs until within-chain variation roughly equals between-chain variation.

```
n <- c(50,5)
sample <- bwalk.metro(ff=nm, init.state=0, run.time=2*n[1], delta=0.8,
theta=20, sample.size=n[2])
sample.discarded <- window(sample,start=n[1]+1,end=2*n[1])
densplot(sample.discarded)
lines(x, nm(x), lty=2, col='red')
```

Between-chain and within-chain variances.

We generate n_2 runs $X_i^{(j)}$, $i = 1, \ldots, n_1$ (after discarding the first half of the simulations). Then we define the **between-chain variance** B and **within-chain variance** W by

$$B/n_1 = \frac{\sum_{j=1}^{n_2}(\bar{X}_\cdot^{(j)} - \bar{X})^2}{n_2 - 1}; \qquad W = \frac{1}{n_2}\sum_{j=1}^{n_2}\left[\frac{\sum_{i=1}^{n_1}(X_i^{(j)} - \bar{X}_\cdot^{(j)})^2}{n_1 - 1}\right]$$

where $\bar{X}_\cdot^{(j)}$ is the average over $X_i^{(j)}$, $i = 1, \ldots, n_1$, and $\bar{X}$ is the overall average. A weighted average

$$\hat{V} = \frac{n_1 - 1}{n_1} W + B/n_1$$

overestimates the variance of target distribution while the within-chain variance W should be an underestimate of variance; here individual runs have not had time to range over all of the target distribution and, therefore, it has less variability.

Potential scale reduction by Gelman and Rubin.

The convergence of X_t's can be detected by the **potential scale reduction factor**

$$\hat{R} = \sqrt{\hat{V}/W}$$

and it decreases to 1 as n_1 goes to the infinity. If the potential scale reduction is high ($\hat{R} \geq 1.1$) then further simulations may improve Monte Carlo integrations for the target distribution (Gelman et al. [1]).

```
source("psrf.r")
cat("Rhat =", psrf(sample.discarded), "")
gelman.diag(sample.discarded, autoburnin=F)
```

Gelman and Rubin [2] proposed an overestimate $\hat{V}$ accounded for the extra variance $B/n_1 n_2$ of the Student's t-distribution. This leads to further modification of $\hat{R}$, which is produced by `gelman.diag()`. It should be noted that one of these factors $\hat{R}$ gets near 1 (say, $\hat{R} = 1.05$) the other is almost identically small.


Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin.
Bayesian data analysis.
Texts in Statistical Science Series. CRC Press, Boca Raton, FL, third edition, 2014.


Andrew Gelman and Donald B Rubin.
Inference from iterative simulation using multiple sequences.
Statistical science, 7(4):457–472, 1992.