STAT 576 Bayesian Analysis

Lecture 7: Bayesian Computation

Chencheng Cai

Washington State University

Monte Carlo Methods

- Suppose we have $x_1, \ldots, x_n$ i.i.d. from a distribution $p(x)$.
- Let $f(x)$ be a measurable function with finite expectation under p.

Monte Carlo Methods

- Suppose we have $x_1, \ldots, x_n$ i.i.d. from a distribution $p(x)$.
- Let $f(x)$ be a measurable function with finite expectation under p.
- By law of large numbers, we have

$$\bar{f}_n = \frac{1}{n}\left(f(x_1) + f(x_2) + \cdots + f(x_n)\right) \xrightarrow{P} \mathbb{E}[f(x)] = \int f(x)p(x)d\mu(x)$$

Monte Carlo Methods

- Suppose we have $x_1, \ldots, x_n$ i.i.d. from a distribution $p(x)$.
- Let $f(x)$ be a measurable function with finite expectation under p.
- By law of large numbers, we have

$$\bar{f}_n = \frac{1}{n}\left(f(x_1) + f(x_2) + \cdots + f(x_n)\right) \xrightarrow{P} \mathbb{E}[f(x)] = \int f(x)p(x)d\mu(x)$$

- By central limit theorem, we have

$$\sqrt{n}\left(\bar{f}_n - \mathbb{E}[f(x)]\right) \xrightarrow{\mathcal{D}} \mathcal{N}(0, \sigma^2),$$

where

$$\sigma^2 = \mathrm{Var}[f(x)] = \int (f(x) - \mathbb{E}[f(x)])^2 p(x)d\mu(x)$$

Monte Carlo Methods

Now we consider the reverse.

- If we want to compute the following integral:

$$I = \int_D f(x) d\mu(x)$$

Monte Carlo Methods

Now we consider the reverse.

- If we want to compute the following integral:

$$I = \int_D f(x) d\mu(x)$$

- Method 1:
  - Generate $x^{(1)}, \dots, x^{(n)}$ i.i.d. and uniformly from D.
  - Estimate the integral by the sample mean:

$$\hat{I}_n = |D| \frac{f(x^{(1)}) + f(x^{(2)}) + \cdots + f(x^{(n)})}{n}$$

Monte Carlo Methods

Now we consider the reverse.

- If we want to compute the following integral:

$$I = \int_D f(x) d\mu(x)$$

- Method 1:
  - Generate $x^{(1)}, \ldots, x^{(n)}$ i.i.d. and uniformly from D.
  - Estimate the integral by the sample mean:

$$\hat{I}_n = |D| \frac{f(x^{(1)}) + f(x^{(2)}) + \cdots + f(x^{(n)})}{n}$$

  - Variance:

$$\mathrm{var}[\hat{I}_n] = \frac{|D|^2}{n} \mathrm{Var}_{\mathsf{unif}}[f(x)] = \frac{|D|^2}{n} \int_D \left(f(x) - \frac{I}{|D|} \right)^2 \frac{1}{|D|} d\mu(x)$$

Monte Carlo Methods

- Method 2:
  - Generate $x^{(1)}, \ldots, x^{(n)}$ i.i.d. from a non-uniform distribution $p(x)$ on D.

Monte Carlo Methods

- Method 2:
  - Generate $x^{(1)}, \ldots, x^{(n)}$ i.i.d. from a non-uniform distribution $p(x)$ on D.
  - Estimate the integral by the sample mean:

$$\hat{I}_n = \frac{1}{n} \sum_{i=1}^{n} \frac{f(x^{(i)})}{p(x^{(i)})}$$

Monte Carlo Methods

- Method 2:
  - Generate $x^{(1)}, \dots, x^{(n)}$ i.i.d. from a non-uniform distribution $p(x)$ on D.
  - Estimate the integral by the sample mean:

$$\hat{I}_n = \frac{1}{n} \sum_{i=1}^{n} \frac{f(x^{(i)})}{p(x^{(i)})}$$

  - Variance:

$$\mathrm{Var}[\hat{I}_n] = \frac{1}{n} \mathrm{Var}_p \left[\frac{f(x)}{p(x)} \right] = \frac{1}{n} \int_D \left(\frac{f(x)}{p(x)} - I \right)^2 p(x) d\mu(x)$$

Monte Carlo Methods

- Method 2:
  - Generate $x^{(1)}, \dots, x^{(n)}$ i.i.d. from a non-uniform distribution $p(x)$ on D.
  - Estimate the integral by the sample mean:

$$\hat{I}_n = \frac{1}{n} \sum_{i=1}^{n} \frac{f(x^{(i)})}{p(x^{(i)})}$$

  - Variance:

$$\mathrm{Var}[\hat{I}_n] = \frac{1}{n} \mathrm{Var}_p \left[\frac{f(x)}{p(x)} \right] = \frac{1}{n} \int_D \left(\frac{f(x)}{p(x)} - I \right)^2 p(x) d\mu(x)$$

  - $p(x)$ is known as the **sampling** distribution.
  - The sampling distribution that minimizes the variance of $\hat{I}_n$ is

$$p(x) \propto f(x)$$

Monte Carlo Methods

$$I = \int_D f(x) d\mu(x)$$

- The optimal sampling distribution is

$$q(x) = \frac{f(x)}{I}$$

Monte Carlo Methods

$$I = \int_D f(x) d\mu(x)$$

- The optimal sampling distribution is

$$q(x) = \frac{f(x)}{I}$$

- For any sampling distribution $p(x)$, we have

$$\mathrm{Var}[\hat{I}_n] = \frac{I^2}{n} \underbrace{\int_D \left(\frac{q(x)}{p(x)} - 1 \right)^2 p(x) d\mu(x)}_{\chi^2\text{-divergence: } \chi^2(q||p)}$$

Monte Carlo Methods

$$I = \int_D f(x) d\mu(x)$$

- The optimal sampling distribution is

$$q(x) = \frac{f(x)}{I}$$

- For any sampling distribution $p(x)$, we have

$$\mathrm{Var}[\hat{I}_n] = \frac{I^2}{n} \underbrace{\int_D \left(\frac{q(x)}{p(x)} - 1 \right)^2 p(x) d\mu(x)}_{\chi^2\text{-divergence: } \chi^2(q||p)}$$

- The variance of the Monte Carlo estimator depends on the χ^2 divergence between the sampling distribution and the optimal one.
- In practice, $q(x)$ is not always tractable. We should choose tractable $p(x)$ that is close to $q(x)$.

Example 1

We want to compute the following integral

$$\int_0^1 (1 - 2|x - 0.5|)\, dx$$

Method 1: draw samples from $\text{unif}[0, 1]$.

```
f <- function(x){1 - 2*abs(x-0.5)}

n = 20
r = 100

Ihat_unif = rep(0, r)
for(i in 1:r){
  x = runif(n)
  Ihat_unif[i] = mean(f(x))
}
```

Example

$$\int_0^1 (1 - 2|x - 0.5|)\, dx$$

Method 1 + vectorization: draw samples from $\mathrm{unif}[0, 1]$.

Example

$$\int_0^1 (1 - 2|x - 0.5|)\, dx$$

Method 1 + vectorization: draw samples from $\mathrm{unif}[0, 1]$.

```
n = 20
r = 100

x = matrix(runif(n*r), ncol = r)
Ihat_unif = colMeans(f(x))
hist(Ihat_unif)
```

- Runtime without vectorization: 0.346 ms
- Runtime with vectorization: 0.025 ms

Example

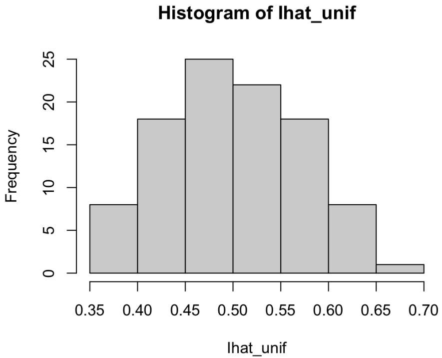
$$\int_0^1 (1 - 2|x - 0.5|)\, dx$$

Method 1 + vectorization: draw samples from $\mathrm{unif}[0, 1]$.

```
n = 20
r = 100

x = matrix(runif(n*r), ncol = r)
Ihat_unif = colMeans(f(x))
hist(Ihat_unif)
```

- Runtime without vectorization: 0.346 ms
- Runtime with vectorization: 0.025 ms



Example

$$\int_0^1 (1 - 2|x - 0.5|)\, dx$$

Method 2: draw samples from $\mathrm{Beta}(2, 2)$.

Example

$$\int_0^1 (1 - 2|x - 0.5|)\, dx$$

Method 2: draw samples from $\mathrm{Beta}(2, 2)$.

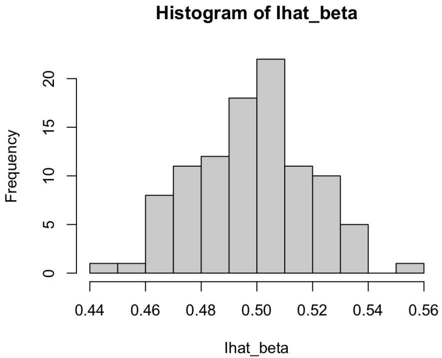
```
x = matrix(rbeta(n*r, 2, 2), ncol=r)
Ihat_beta = colMeans(f(x) / dbeta(x,
    2, 2))
hist(Ihat_beta)
```

Example

$$\int_0^1 (1 - 2|x - 0.5|)\, dx$$

Method 2: draw samples from $\mathrm{Beta}(2, 2)$.

```
x = matrix(rbeta(n*r, 2, 2), ncol=r)
Ihat_beta = colMeans(f(x) / dbeta(x,
     2, 2))
hist(Ihat_beta)
```



Example

$$\int_0^1 (1 - 2|x - 0.5|)\, dx$$

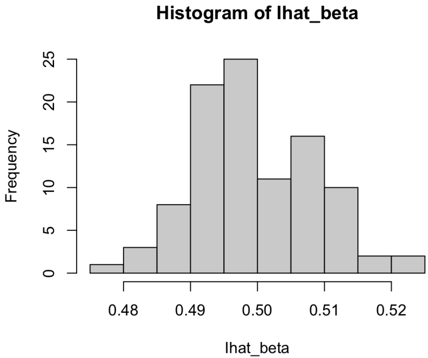
Method 3: draw samples from $\mathrm{Beta}(2, 2)$ with more MC samples.

Example

$$\int_0^1 (1 - 2|x - 0.5|)\, dx$$

Method 3: draw samples from $\mathrm{Beta}(2, 2)$ with more MC samples.

```
n = 100
x = matrix(rbeta(n*r, 2, 2), ncol=r)
Ihat_beta = colMeans(f(x) / dbeta(x,
    2, 2))
hist(Ihat_beta)
```



Quasi Monte Carlo Methods

- Monte Carlo method: draw $x^{(1)}, \ldots, x^{(n)}$ i.i.d. from a sampling distribution.
- Quasi Monte Carlo method: pick $x^{(1)}, \ldots, x^{(n)}$ to represent the sampling distribution.

Quasi Monte Carlo Methods

- Monte Carlo method: draw $x^{(1)}, \ldots, x^{(n)}$ i.i.d. from a sampling distribution.
- Quasi Monte Carlo method: pick $x^{(1)}, \ldots, x^{(n)}$ to represent the sampling distribution.
- The samples in the quasi Monte Carlo method are deterministic and are assume to be "uniform" in the whole space.
- The sample sequence $x^{(1)}, x^{(2)}, \ldots$ is called **low discrepancy sequence** (e.g. Sobel sequence).

Example

$$\int_0^1 (1 - 2|x - 0.5|)\, dx$$

Method 4: QMC samples from $\mathrm{unif}[0,1]$.

Example

$$\int_0^1 (1 - 2|x - 0.5|)\, dx$$

Method 4: QMC samples from $\mathrm{unif}[0, 1]$.

```
x = (seq(n)-0.5)/n
Ihat_unif_qmc = mean(f(x))
print(Ihat_unif_qmc)
```

The outcome is 0.5.

Example

$$\int_0^1 (1 - 2|x - 0.5|)\, dx$$

Method 5: QMC samples from $\mathrm{Beta}(2,2)$.

Example

$$\int_0^1 (1 - 2|x - 0.5|)\, dx$$

Method 5: QMC samples from $\mathrm{Beta}(2, 2)$.

```
x = (seq(n)-0.5)/n
y = qbeta(x, 2, 2)
Ihat_beta_qmc = mean(f(y)/dbeta(y, 2, 2))
print(Ihat_beta_qmc)
```

The outcome is 0.50002.

Random Number Generator

- Most currently used random number generators on modern computers are **pseudo random number generators (PRNG)**.

Random Number Generator

- Most currently used random number generators on modern computers are **pseudo random number generators (PRNG)**.
- PRNG is a deterministic sequence that requires a starting value (known as **seed**).

Random Number Generator

- Most currently used random number generators on modern computers are **pseudo random number generators (PRNG)**.
- PRNG is a deterministic sequence that requires a starting value (known as **seed**).
- The sequence generated by PRNG behaves like independent random numbers.
- The sequence generated by PRNG will finally repeat.
- Two sequences generated by the same PRNG and the same seed should be identical.

Random Number Generator

- Most currently used random number generators on modern computers are **pseudo random number generators (PRNG)**.
- PRNG is a deterministic sequence that requires a starting value (known as **seed**).
- The sequence generated by PRNG behaves like independent random numbers.
- The sequence generated by PRNG will finally repeat.
- Two sequences generated by the same PRNG and the same seed should be identical.
- Common practices:
  - Set the seed at the beginning of your program for easy replication of the results.

    ```
    set.seed(0)
    ```

  - Do not abuse it! Use a predetermined seed instead of optimizing it.

Generating Random Numbers

- The default random numbers generated by PRNG are i.i.d. $\mathrm{unif}[0, 1]$.
- How do we generate random numbers from an arbitrary univariate distribution F?

Generating Random Numbers

- The default random numbers generated by PRNG are i.i.d. $\mathrm{unif}[0, 1]$.
- How do we generate random numbers from an arbitrary univariate distribution F?
  - Transformation.
  - Inverse C.D.F.
  - Accept-reject sampling.

Generating Random Numbers — Transformation

Let $u_1, u_2, \ldots$ be a sequence of i.i.d. $\mathrm{unif}[0,1]$ random variables.

Generating Random Numbers — Transformation

Let $u_1, u_2, \ldots$ be a sequence of i.i.d. $\mathrm{unif}[0,1]$ random variables.

- Let $z_i = \mathbb{I}\{u_i > 0.5\}$.
  Then $z_1, z_2, \ldots$ is an i.i.d. sequence of $\mathrm{Bernoulli}(0.5)$ random variables.

Generating Random Numbers — Transformation

Let $u_1, u_2, \ldots$ be a sequence of i.i.d. $\text{unif}[0,1]$ random variables.

- Let $z_i = \mathbb{I}\{u_i > 0.5\}$.
  Then $z_1, z_2, \ldots$ is an i.i.d. sequence of $\text{Bernoulli}(0.5)$ random variables.
- Let $y_i = \sum_{j=1}^{n} z_{n(i-1)+j}$.
  Then $y_1, y_2, \ldots$ is an i.i.d. sequence of $\text{Binomial}(n, 0.5)$ random variables.

Generating Random Numbers — Transformation

Let $u_1, u_2, \ldots$ be a sequence of i.i.d. $\mathrm{unif}[0,1]$ random variables.

- Let $z_i = \mathbb{I}\{u_i > 0.5\}$.
  Then $z_1, z_2, \ldots$ is an i.i.d. sequence of $\mathrm{Bernoulli}(0.5)$ random variables.
- Let $y_i = \sum_{j=1}^{n} z_{n(i-1)+j}$.
  Then $y_1, y_2, \ldots$ is an i.i.d. sequence of $\mathrm{Binomial}(n, 0.5)$ random variables.
- Let $d_i^j = \lfloor 2^j u_i \rfloor \mod 2$. That is $u_i = 0.d_i^1 d_i^2 d_i^3 \ldots$ is a base-2 representation.
  Then d_i^j's are i.i.d. $\mathrm{Bernoulli}(0.5)$.

Generating Random Numbers — Transformation

Let $u_1, u_2, \ldots$ be a sequence of i.i.d. $\mathrm{unif}[0,1]$ random variables.

- Let $z_i = \mathbb{I}\{u_i > 0.5\}$.
  Then $z_1, z_2, \ldots$ is an i.i.d. sequence of $\mathrm{Bernoulli}(0.5)$ random variables.
- Let $y_i = \sum_{j=1}^{n} z_{n(i-1)+j}$.
  Then $y_1, y_2, \ldots$ is an i.i.d. sequence of $\mathrm{Binomial}(n, 0.5)$ random variables.
- Let $d_i^j = \lfloor 2^j u_i \rfloor \mod 2$. That is $u_i = 0.d_i^1 d_i^2 d_i^3 \ldots$ is a base-2 representation.
  Then d_i^j's are i.i.d. $\mathrm{Bernoulli}(0.5)$.
- Let $x_i = \sum_{j=1}^{n} d_i^j$.
  Then $x_1, x_2, \ldots$ is an i.i.d. sequence of $\mathrm{Binomial}(n, 0.5)$ random variables.

Generating Random Numbers — Transformation

Let $u_1, u_2, \ldots$ be a sequence of i.i.d. $\mathrm{unif}[0,1]$ random variables.

- Let $z_i = \mathbb{I}\{u_i > 0.5\}$.
  Then $z_1, z_2, \ldots$ is an i.i.d. sequence of $\mathrm{Bernoulli}(0.5)$ random variables.
- Let $y_i = \sum_{j=1}^{n} z_{n(i-1)+j}$.
  Then $y_1, y_2, \ldots$ is an i.i.d. sequence of $\mathrm{Binomial}(n, 0.5)$ random variables.
- Let $d_i^j = \lfloor 2^j u_i \rfloor \mod 2$. That is $u_i = 0.d_i^1 d_i^2 d_i^3 \ldots$ is a base-2 representation.
  Then d_i^j's are i.i.d. $\mathrm{Bernoulli}(0.5)$.
- Let $x_i = \sum_{j=1}^{n} d_i^j$.
  Then $x_1, x_2, \ldots$ is an i.i.d. sequence of $\mathrm{Binomial}(n, 0.5)$ random variables.
- Let $w_i = 2u_i$.
  Then $w_1, w_2, \ldots$ is an i.i.d. sequence of $\mathrm{unif}[0,2]$ random variables.

Generating Random Numbers — Transformation

Let $u_1, u_2, \ldots$ be a sequence of i.i.d. $\text{unif}[0,1]$ random variables.

- Let $z_i = \mathbb{I}\{u_i > 0.5\}$.
  Then $z_1, z_2, \ldots$ is an i.i.d. sequence of $\text{Bernoulli}(0.5)$ random variables.
- Let $y_i = \sum_{j=1}^{n} z_{n(i-1)+j}$.
  Then $y_1, y_2, \ldots$ is an i.i.d. sequence of $\text{Binomial}(n, 0.5)$ random variables.
- Let $d_i^j = \lfloor 2^j u_i \rfloor \mod 2$. That is $u_i = 0.d_i^1 d_i^2 d_i^3 \ldots$ is a base-2 representation.
  Then d_i^j's are i.i.d. $\text{Bernoulli}(0.5)$.
- Let $x_i = \sum_{j=1}^{n} d_i^j$.
  Then $x_1, x_2, \ldots$ is an i.i.d. sequence of $\text{Binomial}(n, 0.5)$ random variables.
- Let $w_i = 2u_i$.
  Then $w_1, w_2, \ldots$ is an i.i.d. sequence of $\text{unif}[0,2]$ random variables.
- Let $r_i = -\log u_i$.
  Then $r_1, r_2, \ldots$ is an i.i.d. sequence of $\text{Exp}(1)$ random variables.

Generating Random Numbers — Inverse C.D.F.

A special type of transformation is using the inverse c.d.f. function.

Generating Random Numbers — Inverse C.D.F.

A special type of transformation is using the inverse c.d.f. function.

- The c.d.f. of a distribution F is given by

$$F(x_0) = \mathbb{P}[x \leq x_0]$$

- The inverse c.d.f. is given by

$$F^{-1}(q) = \inf \{x : F(x) \geq q\}$$

Generating Random Numbers — Inverse C.D.F.

A special type of transformation is using the inverse c.d.f. function.

- The c.d.f. of a distribution F is given by

$$F(x_0) = \mathbb{P}[x \leq x_0]$$

- The inverse c.d.f. is given by

$$F^{-1}(q) = \inf \{x : F(x) \geq q\}$$

- If $u_1, u_2, \ldots$ is an i.i.d. sequence of $\mathrm{unif}[0,1]$ random variables, then $F^{-1}(u_1), F^{-1}(u_2), \ldots$ is an i.i.d. sequence of F random variables.

Generating Random Numbers — Inverse C.D.F.

A special type of transformation is using the inverse c.d.f. function.

- The c.d.f. of a distribution F is given by

$$F(x_0) = \mathbb{P}[x \leq x_0]$$

- The inverse c.d.f. is given by

$$F^{-1}(q) = \inf \{x : F(x) \geq q\}$$

- If $u_1, u_2, \ldots$ is an i.i.d. sequence of $\mathrm{unif}[0, 1]$ random variables, then $F^{-1}(u_1), F^{-1}(u_2), \ldots$ is an i.i.d. sequence of F random variables.
- **Justification:**

$$\mathbb{P}[F^{-1}(u_1) \leq x_0] = \mathbb{P}[u_1 \leq F(x_0)] = F(x_0)$$

Example: Generating Standard Normal Random Variables

Method 1: approximated inverse c.d.f.

Example: Generating Standard Normal Random Variables

Method 1: approximated inverse c.d.f.
We approximate the inverse c.d.f. of a standard normal by (for $0 < q < 1/2$)

$$\Phi^{-1}(q) \approx t - \frac{c_0 + c_1 t + c_2 t^2}{1 + d_1 t + d_2 t^2 + d_3 t^3}$$

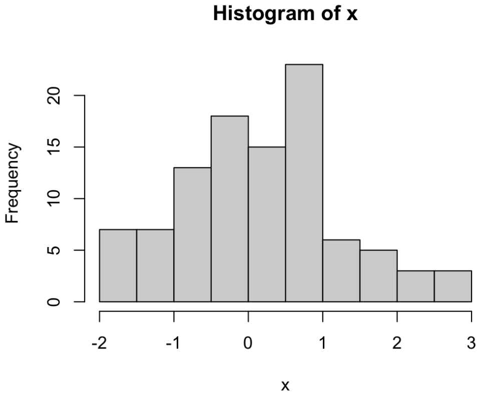
for $t = \sqrt{-2 \log q}$ and

$$\begin{aligned} c_0 &= 2.515517 & \qquad d_1 &= 1.432788 \\ c_1 &= 0.802853 & d_2 &= 0.189269 \\ c_2 &= 0.010328 & d_3 &= 0.001308 \end{aligned}$$

Example: Generating Standard Normal Random Variables

```
c0 = 2.515517
c1 = 0.802853
c2 = 0.010328
d1 = 1.432788
d2 = 0.189269
d3 = 0.001308

u = runif(100)
t = sqrt(-2*log(abs(u-0.5)))
denum = c0 + c1*t + c2*t**2
num = 1 + d1*t + d2*t**2 + d3*t**3
x = t - denum/num
x = x * sign(u - 0.5)
hist(x)
```



Example: Generating Standard Normal Random Variables

Method 2: Box-Muller transformation.

- Assume x_1 and x_2 are independent standard normal random variables.
- The joint density is

$$p(x_1, x_2) \propto e^{-\frac{x_1^2 + x_2^2}{2}}$$

Example: Generating Standard Normal Random Variables

Method 2: Box-Muller transformation.

- Assume x_1 and x_2 are independent standard normal random variables.
- The joint density is

$$p(x_1, x_2) \propto e^{-\frac{x_1^2 + x_2^2}{2}}$$

- Consider the following transformation

$$r = \sqrt{x_1^2 + x_2^2} \qquad x_1 = r\cos\theta$$

$$\theta = \arctan\frac{x_2}{x_1} \qquad x_2 = r\sin\theta$$

Example: Generating Standard Normal Random Variables

Method 2: Box-Muller transformation.

- Assume x_1 and x_2 are independent standard normal random variables.
- The joint density is

$$p(x_1, x_2) \propto e^{-\frac{x_1^2 + x_2^2}{2}}$$

- Consider the following transformation

$$r = \sqrt{x_1^2 + x_2^2} \qquad x_1 = r\cos\theta$$

$$\theta = \arctan\frac{x_2}{x_1} \qquad x_2 = r\sin\theta$$

- The density for (r, θ) is

$$p(r, \theta) = p(x_1, x_2)\left|\frac{\partial(x_1, x_2)}{\partial(r, \theta)}\right| \propto re^{-r^2/2}$$

Example: Generating Standard Normal Random Variables

Method 2: Box-Muller transformation.

- Assume x_1 and x_2 are independent standard normal random variables.
- The joint density is

$$p(x_1, x_2) \propto e^{-\frac{x_1^2 + x_2^2}{2}}$$

- Consider the following transformation

$$r = \sqrt{x_1^2 + x_2^2} \qquad x_1 = r\cos\theta$$

$$\theta = \arctan\frac{x_2}{x_1} \qquad x_2 = r\sin\theta$$

- The density for (r, θ) is

$$p(r, \theta) = p(x_1, x_2)\left|\frac{\partial(x_1, x_2)}{\partial(r, \theta)}\right| \propto re^{-r^2/2}$$

- $\theta \sim \mathrm{unif}[0, 2\pi)$ and $p(r) \propto re^{-r^2/2}$ with c.d.f. $1 - e^{-r^2/2}$ (i.e. $r^2 \sim \mathrm{Exp}(1/2)$)

Example: Generating Standard Normal Random Variables

```
u = runif(100)
theta = runif(100) * 2 * pi
r = sqrt(-2*log(u))
x1 = r * sin(theta)
x2 = r * cos(theta)
x = c(x1, x2)
hist(x)
```