2018 Fall

Computational Statistics
HW#7

182STG18 이하경

Ⅰ. Description

Approximate Simulation

- Sampling Importance Resampling (SIR)

HW6 에서 다루었던 Rejection Sampling 은 Exact Simulation 의 일종이다. target distribution 인 f 를 따르는 random variable 들의 iid sample 을 생성하고자 할 때 f 함수를 알고 있으나 sampling 을 직접적으로 수행하기 어려운 경우 좀 더 쉽게 sampling 을 가능하게 하는 방법으로, 추출된 최종 sample 은 실제 f 를 완벽히 따르지만, 초기 표본에서 확률적으로 생성되어 정해진 크기(n)의 표본을 얻고자 할 경우 필요한 초기 sample 의 수를 알 수 없다(random)는 특징이 있다.

Exact Sampling 과 다르게 Approximate Simulation 은 좀 더 쉬운 근사적인 방법으로 Sampling 을 진행하지만 실제 target f 를 근사적으로 따른다. 그 중 하나로 Sampling Importance Resampling (SIR)은 다음의 순서에 따라 진행된다.

1) Importance Sampling Function(또는 envelope) 'g'로부터 m 개의 $Y_1, \dots, Y_m$ iid sample을 추출한다.
2) Sample 의 각 값들을 이용해 weight $w(Y_i)$을 계산한다.

$$w(Y_i) = \frac{f(Y_i)/g(Y_i)}{\sum_{j=1}^{m} f(Y_j)/g(Y_j)} : \ standardized\ importance\ weight, \ \sum_{i=1}^{n} w(Y_i) = 1$$

3) (1)의 m 개의 초기 sample 로부터 resampling probability $w(Y_1), \dots, w(Y_m)$ 을 이용한 비복원추출로 n 개의 최종 sample 을 생성한다.

이렇게 생성된 크기 n 의 sample 은 근사적으로 target distribution f 를 따르며, 초기 표본의 수 m 이 클수록 실제 분포와 가까워진다. target f 와 envelope 로 설정하는 g 가 가까울 수록 resampling probability 가 커진다는 점을 기반으로 하여, 각 표본 값에서의 sampling probability 는 실제 density f 와 연결되므로 생성된 sample 을 f 분포 하의 추정에 합리적으로 이용할 수 있다는 데 의의가 있다.

SIR 은 Rejection Sampling 과 다르게 초기 m 개의 표본으로부터 얻고자 하는 표본의 크기 n 을 사전에 정할 수 있지만 실제 f 를 근사적으로 따른다는 특징이 있다. SIR 에서는 f 분포로의 수렴을 위해 m 과 n 의 적절한 비율로 설정한다. 일반적으로 고정된 m 값에서 n 은 작아야 하나, 추정 시 분산을 줄이기 위해 n 은 일정 크기 이상이 필요하다는 점도 고려해야 한다.

Implementation 1 과 2 에서는 SIR 을 이용해 sample 을 직접 추출해보고, 특히 Implementation 2 에서는 더 나아가 m 과 n 의 비율에 따른 결과 확인과 Rejection Sampling 결과와의 비교를 다룬다.

Variance Reduction Techniques

- Importance Sampling (IS)

$E[h(X)] = \int h(x)f(x)dx$ 을 추정하고자 할 때, Monte-Carlo estimator $\hat{\mu}_{MC}$ 는 f(x)로부터 위의 Rejection Sampling 이나 SIR 과 같은 방법을 적절히 이용해 생성된 n 개의 iid random sample $X_1, \dots, X_n$ 로부터

$$\hat{\mu}_{MC} = \frac{1}{n}\sum_{i=1}^{n} h(X_i)$$

와 같이 단순하게 계산할 수 있다. 그러나 이보다 분산이 더 작은 정밀한 추정치를 Importance Sampling Approach 를 통해 얻을 수 있다. Importance Sampling(IS)이란 관심있는 event 에 대해 가중치를 크게 두고 oversampling 한 후 Importance Weights(IW)를 사용해 bias 를 조정하여 더 정밀한 추정을 가능하게 하는 방법으로, 관심 event 가 발생할 확률이 매우 작은 경우에 분산을 크게 줄여 유용하게 사용할 수 있다.

$$\mu = \int h(x)f(x)dx = \int h(x)\frac{f(x)}{g(x)}g(x)dx = \int h(x)w^*(x)\,g(x)dx$$

위의 식처럼 기댓값 μ 를 target f 대신 다른 density g 를 이용하여 추정할 수 있으며 이 때 g 함수를 Importance Sampling Function(ISF) 라고 한다. IS 추정치는 importance weight 에 따라 다음과 같이 계산할 수 있다.

$$\hat{\mu}_{IS}^{*} = \frac{1}{n}\sum_{i=1}^{n} h(X_i)w^*(X_i), \qquad w^*: unstandardized\ importance\ weights$$

$$\hat{\mu}_{IS} = \sum_{i=1}^{n} h(X_i)w(X_i), \qquad w: standardized\ importance\ weights$$

여기서 $\hat{\mu}_{IS}^{*}$은 unbiased estimator 이지만 $\hat{\mu}_{IS}$은 약간의 bias 가 존재한다. Standardized IW 를 사용하는 $\hat{\mu}_{IS}$은 target distribution f 를 고려할 때 비례상수 값을 알지 않아도 되는 Bayesian Posterior 와 같은 경우에 특히 유용하다. IS Approach 를 통한 추정치는 sampling 시 standardized IW 를 사용하는 SIR 방법으로 생성한 iid sample $X_1, \ldots, X_n$ 으로부터 단순추정한 $\hat{\mu}_{SIR} = \frac{1}{n}\sum_{i=1}^{n} h(X_i)$와 비교했을 때 다음과 같은 성질을 가진다.

$$\mathrm{E}(\hat{\mu}_{SIR}) = \mathrm{E}(\hat{\mu}_{IS})$$

$$\mathrm{Var}(\hat{\mu}_{SIR}) \geq \mathrm{Var}(\hat{\mu}_{IS})$$

따라서 sample 로부터 기댓값 μ 를 추정하고자 하는 경우에 IS 추정치는 정밀성 측면에서 더 좋은 추정치라고 할 수 있다.

Implementation 3 에서는 Network Failure Example 에서 기댓값 μ의 추정치와 분산의 추정치를 Simple MC Approach 와 IS Approach 를 이용해 계산해보고, 방법에 따른 차이를 비교해본다.

Ⅱ. Implementation

1. SIR of Slash Distribution (Example 6.3)

Goal

$$Y = X/U,\ \ X \sim N(0,1)\ \&\ U \sim U(0,1)\ (indep.) \ \ \rightarrow \ \ Y \sim Slash\ Dist^{n}$$

1) SIR Algorithm 을 이용해 Importance Sampling Function g 를 slash distribution 으로 하여 target distribution standard normal 의 sample 을 생성한다.

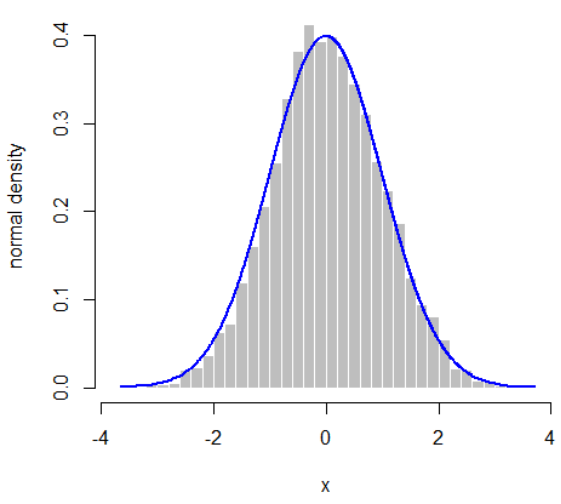
2) SIR Algorithm 을 이용해 ISF g 를 standard normal distribution 으로 하여 target distribution slash 의 sample 을 생성한다.

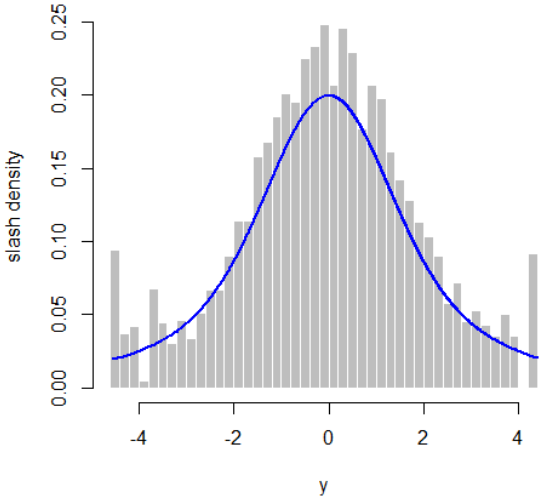
위의 두 가지 Sampling 을 진행하여 Histogram 을 통해 적절한지 확인하고, 결과를 비교한다.

초기 sample size m=100,000, resample size n=5,000 을 이용하였다. $(n/m = 0.05)$

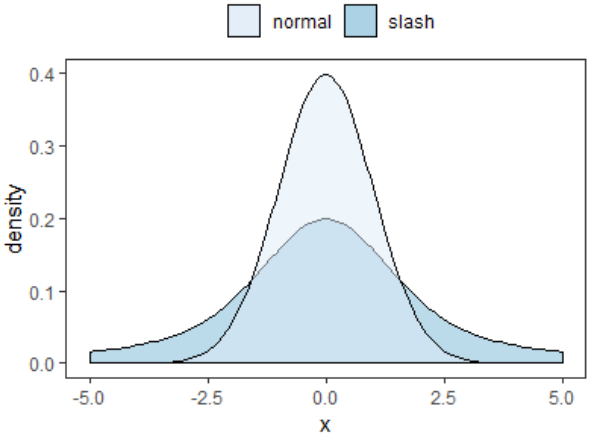
Result

target	ISF (envelope)	target f	ISF (envelope)
standard normal	slash	slash	standard normal





각 분포에서 생성된 sample 의 실제 히스토그램과, target distribution 의 density line 을 겹쳐 그린 결과이다. 먼저 slash 분포를 envelope 로 사용하여 5000 개의 normal sample 을 생성한 경우 실제 target 분포와 거의 동일한 경향을 보이는 것을 확인할 수 있다. 반면 표준정규분포를 ISF 로 사용한 경우 생성된 sample 은 그래프에서 확인할 수 있듯이 target 인 slash 분포를 잘 따르고 있다고 할 수 없다. 특히 꼬리부분의 값들이 실제 density 보다 더 높은 빈도를 보이고 있는데, 오른쪽에서 표준 정규분포와 slash 분포의 density 를 비교해보면 양쪽 끝에서 slash 분포의 꼬리가 더 두꺼워 normal 분포에 좋은 envelope 가 될 수 있지만 반대의 경우 $(f = \text{slash}, g = \text{std. normal})$ f/g가 상대적으로 커져, 기대하지 않는 값들이 sample 에 포함될 가능성이 커지는 것이 원인이라고 할 수 있다.



2. Bayesian Inference (Example 6.4)

Obs	(8, 3, 4, 3, 1, 7, 2, 6, 2, 7)
Prior $f(\lambda)$	$\log\lambda \sim \mathrm{N}(\log 4, 0.5^2)$
Likelihood $L(\lambda\|X)$	$X \mid \lambda \sim Poission(\lambda)$
Unnormalized Posterior	$q(\lambda\|X) \propto f(\lambda) \cdot L(\lambda\|X)$

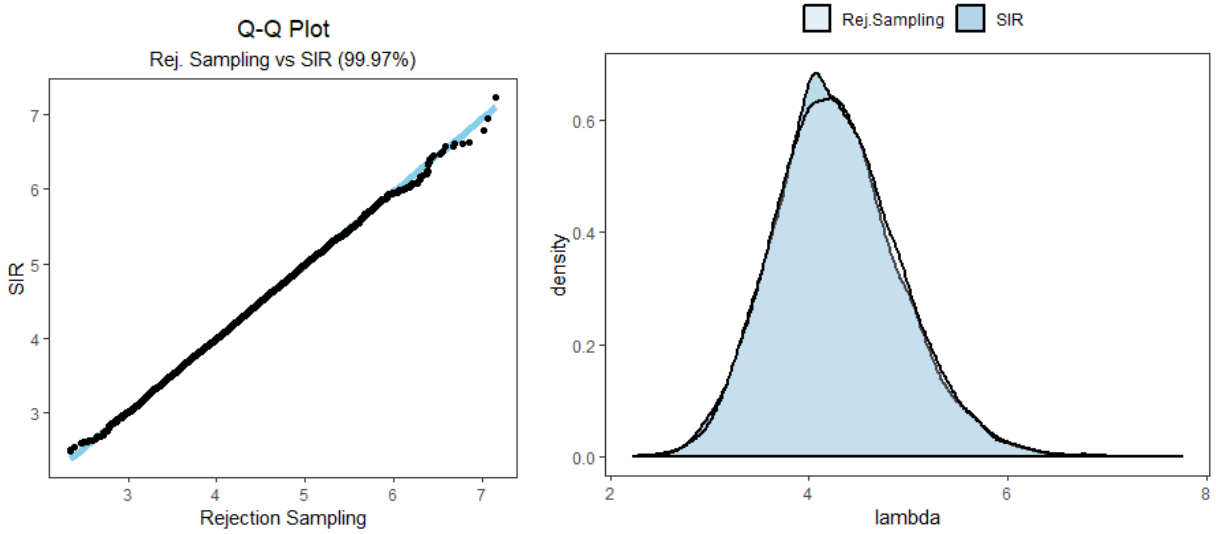
Goal

$\boldsymbol{\lambda}$ 의 사후분포로부터 sample 을 SIR 을 이용해 생성해보고 HW6 의 Rejection Sampling 에서 생성된 sample 과 Q-Q Plot 을 그려 결과를 확인한다. 또한 sampling size m 과 n 의 크기 비율에 따른 결과의 차이가 있는지 추가적으로 확인해본다.

Algorithm

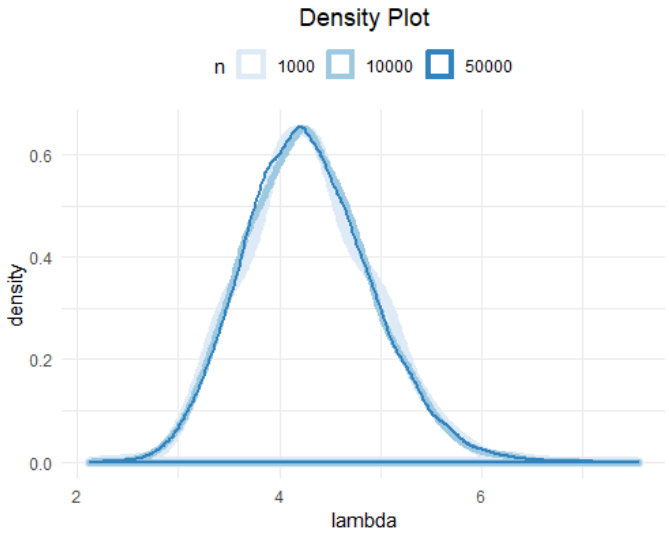
1) log-normal Prior f(λ) 로부터 $\lambda_1, \ldots, \lambda_m$ 개의 초기 sample 을 생성한다.
2) unstandardized weight $\mathrm{w}^*(\lambda_i) = L(\lambda_i|X)$ 를 계산한다.
3) 계산한 가중치를 이용해 m 개의 초기 sample 에서 n 개의 sample 을 복원추출로 다시 생성한다.

Result 1. Comparison between SIR & Rejection Sampling



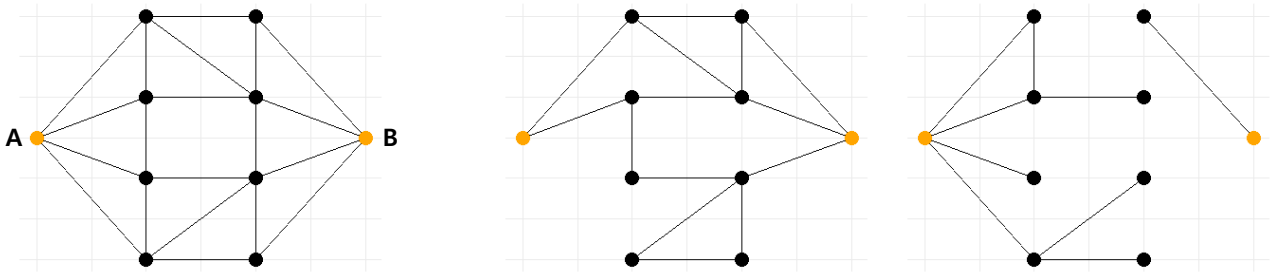
Approximate Simulation 인 SIR 방법에서 초기 size m=100000, resample size n=10000 으로 구성한 sample 과 Exact Simulation 인 Rejection Sampling 에서 n=100000 개 중 acceptance rate=29.05%로 선택된 29052 개의 sample 을 이용하여 Q-Q Plot 과 Density Plot 을 그려본 결과이다. 그래프에서 두 가지 sample 이 거의 동일한 경향을 보이고 있다. Exact Simulation 으로 구성된 sample 은 정확히 target distribution f 를 따르며, SIR 에서 구성된 sample 이 이와 매우 유사하므로 SIR 을 이용한 근사적 방법이 타당하다고 할 수 있다.

Result 2. Changes of resample size n under fixed m=100000



초기 설정 m=100000, n=10000 에서 m 과 n 의 변화에 따른 차이가 있는지 확인하기 위해 고정된 m 에서 n 을 5000 $(n/m = 0.05)$, 10000 $(n/m = 0.1)$, 50000 $(n/m = 0.5)$으로 바꿔가며 density 그래프를 확인해보았다. 이 예제의 경우 비율에 따라 결과에 큰 차이를 보이지는 않았다.

3. Network Failure Probability (Example 6.9)



full-connected **connected** **disconnected**

Goal

A 와 B 사이의 Network 가 위와 같이 20 개의 edge 로 이루어져 있을 때, Network Connection 을 다음과 같이 표현할 수 있으며 각 edge 가 끊어질 확률 p 는 일반적으로 매우 작다.

$$X = (X_1, \dots, X_{20}), \text{ each edge } X_k = \begin{cases} 1 & intact\ edge \\ 0 & failed\ edge' \end{cases}$$

$$b(X) = \#\ of\ broken\ edges\ \ \&\ \ h(X) = \begin{cases} 1 & if\ A\ is\ not\ connected\ to\ B \\ 0 & if\ A\ is\ connected\ to\ B \end{cases}$$

이 경우 기댓값 $\mu = E[h(X)] = P\{h(X) = 1\}$, 즉 A 와 B 가 연결될 확률을 추정하려고 한다.

1) p=0.05 일 때 naïve Monte-Carlo Estimator $\hat{\mu}_{MC} = \frac{1}{n}\sum_{i=1}^{n} h(X_i)$을 구한다.

2) Importance Sampling 을 이용하여 $p^*(> p) = 0.2$일 때 (unstandardized) IS Estimator $\hat{\mu}_{IS}^* = \frac{1}{n}\sum_{i=1}^{n} h(X_i)w^*(X_i)$ 와 (standardized) IS Estimator $\hat{\mu}_{IS} = \sum_{i=1}^{n} h(X_i)w(X_i)$을 구한다.

위 추정치들의 값과 각각의 분산 및 표준오차를 계산하여 방법 별로 비교하고, IS 추정 시 $p^* = 0.2$ 외에 분산을 더 작게 할 수 있는 p^* 값이 있는지 추가적인 시행을 통해 확인해본다.

$$w^*(X_i^*) = \left(\frac{1-p}{1-p^*}\right)^{20}\left[\frac{p(1-p^*)}{p^*(1-p)}\right]^{b(Xi^*)}, \quad X_i^* = \text{each edge of a sample simulated with probability } p^*$$

$Note)\ E\{\hat{\mu}_{MC}\} = \mu,\ Var\{\hat{\mu}_{MC}\} = \frac{1}{n}Var\{h(X)\}$

$$E\{\hat{\mu}_{IS}^*\} = \mu\ (unbiased),\ Var\{\hat{\mu}_{IS}^*\} = \frac{1}{n}Var\{h(X)w^*(X)\}$$

$$E\{\hat{\mu}_{IS}\} = \mu - \frac{1}{n}Cov\{h(X)w^*(X)\} + \frac{\mu}{n}Var\{w^*(X)\} + \mathcal{O}\left(\frac{1}{n^2}\right),$$
$$Var\{\hat{\mu}_{IS}\} = \frac{1}{n}[Var\{h(X)w^*(X)\} + \mu^2 Var\{w^*(X)\} - 2\mu\, Cov\{h(X)w^*(X), w^*(X)\}] + \mathcal{O}\left(\frac{1}{n^2}\right)$$

sample size n=100000 을 이용하였다.

Result 1. Values of Estimation **(단위=10^-5)**

Estimated	naive MC	unstandardized IS	standardized IS
Expected Value	2.000	1.028	1.024
SE	1.414	0.158	0.159
sum of h(X)	2	464	

IS Approach 를 이용해 발생확률 p 를 보다 큰 p*로 대체하여 n 개의 sample 을 구성한 후 weight 를 이용해 조정하여 h(X) 추정치를 계산하였다. 기댓값의 MC 추정치와 unstandardized IS 추정치는 unbiased estimator 이며, standardized IS 추정치는 실제 μ와 비교해 약간의 bias 가 있을 수 있다. 그러나 두가지 IS 추정치의 분산이 단순 계산한 MC 추정치의 분산보다 10 배 가까이 작음을 확인하였다.

Result 2. Standard Errors by different p*'s **(단위=10^-5)**

p*	$\hat{\mu}_{IS}^*$	SE
0.1	1.269	0.321
0.2	**1.028**	**0.158**
0.4	1.141	0.392
0.5	1.103	0.343

p*의 값을 바꿔가면서 추정치의 분산 및 표준오차를 확인했을 때 p*=0.2 보다 분산을 작게 하는 값은 발견되지 않았다. 그러나 모든 경우 p=0.05 로 단순 계산한 MC 추정치의 분산보다 작다.

Ⅲ. Discussion

Exact Simulation 인 Rejection Sampling 의 경우, 특정 크기의 sample 을 얻고자 할 때 초기 sample 에서의 acceptance rate 를 미리 알 수 없으므로 매우 많은 수의 초기 sample 이 필요하거나 여러 번 sampling 을 시행해야 할 수 있다. Approximate Simulation 인 SIR method 는 초기 m 개의 sample 로부터 필요한 n 개의 sample 을 재생성하여 원하는 크기의 sample 을 얻을 수 있으며 뽑힌 값들이 근사적이지만 실제 분포를 거의 유사하게 따르는 것을 예제에서 히스토그램, Q-Q Plot 등으로 직접 확인할 수 있었다. 그러나 이를 위해서는 Importance Sampling Function g 를 적절히 잘 선택하는 것이 매우 중요하며 다음의 조건들을 만족해야 한다.

1. g 의 support 는 target f 의 전체 support 를 포함해야 한다.
2. g 는 f 보다 두꺼운 꼬리를 가져야 한다.

만약 f 의 값이 작은 꼬리 부분에서 g 의 값이 0 에 가까울 경우, weight 를 계산할 때 f/g 값이 매우 커져 sample 에 포함되지 않기를 기대하는 값들이 초기 sampling 시 뽑힐 확률 자체는 희박하지만 초기 sample 에 포함될 경우 매우 큰 가중치로 인해 resampling 시에 많이 포함되는 일이 발생한다. Implementation 1 에서 slash sample 를 얻기 위해 standard normal function 을 envelope 를 사용한 경우에서 이를 확인할 수 있었다.

또한 발생확률 p 가 매우 작은 event 에 대해 기댓값을 추정할 때 IS Approach 를 이용해 단순 추정보다 분산을 크게 줄여 정밀성을 높일 수 있음을 Implementation 3 의 예제를 통해서 알게 되었다.

[Appendix] R Code

 # 1. slash distribution

```
# target f: normal, envelope g: slash
  ym <- rslash(m)
  usiw <- dnorm(ym) / dslash(ym) ; siw <- usiw / sum(usiw)
  xn <- sample(ym, n, replace = T, prob = siw)

  hist(xn, freq = F, breaks = 50, col = "gray", border = "white",
       xlab = "x", ylab = "normal density", main = "generate normal from slash")
  lines(sort(xn), dnorm(sort(xn)), col = "blue", lwd = 2)

# target f: slash, envelope g: normal
  ym <- rnorm(m)
  usiw <- dslash(ym) / dnorm(ym) ; siw <- usiw / sum(usiw)
  xn <- sample(ym, n, replace = T, prob = siw)

  hist(xn, freq = F, breaks = 50, col = "gray", border = "white",
       xlab = "y", ylab = "slash density", main = "generate slash from normal")
  lines(sort(xn), dslash(sort(xn)), col = "blue", lwd = 2)

  gg <- data.frame(x = seq(-5, 5, length = 100)) %>% mutate(slash = dslash(x), normal = dnorm(x))
  ggplot(gg) + theme_test() +
    geom_area(aes(x, slash, fill = "slash"), color = "black", alpha = 0.7) +
    geom_area(aes(x, normal, fill = "normal"), color = "black", alpha = 0.5) +
    scale_fill_brewer("", palette = "Blues") + labs(y = "density") + theme(legend.position = "top")
```

 # 2. Bayesian Inference

```
  data2 <- c(8, 3, 4, 3, 1, 7, 2, 6, 2, 7)
  Lfunc <- function(lambda) {
    lik = c() ;for (i in 1:length(lambda)) lik[i] <- prod(dpois(data2, lambda[i]))
    return(lik)
  }

  myfunc1 <- function(m, n) {
    l <- rlnorm(m, log(4), 0.5)
    usiw <- Lfunc(l)
    l.re <- sample(l, n, replace = T, prob = usiw)
    return(l.re)
  }

  myfunc2 <- function(n) {
    l <- rlnorm(n, log(4), 0.5) ; u <- runif(n)
    l.keep <- l[(u <= Lfunc(l)/Lfunc(4.3))]
    return(l.keep)
  }

  m = 100000 ; n = 10000
  l.sir <- myfunc1(m, n)
  l.rs <- myfunc2(m)
  l.rs.n <- sample(l.rs, n)

  l.qq <- data.frame(l.rs = sort(l.rs.n), l.sir = sort(l.sir))
  ggplot(l.qq) +
    theme_test() + theme(plot.title = eleyment_text(hjust = 0.5), plot.subtitle = element_text(hjust = 0.5)) +
    geom_smooth(aes(l.rs, l.sir), size = 2, color = "skyblue", method = "lm") + geom_point(aes(l.rs, l.sir)) +
```

```
    labs(title = "Q-Q Plot",
         subtitle = paste("Rej. Sampling vs SIR (", round(summary(lm(l.sir ~ l.rs, l.qq))$r.squared, 4)*100, "%)", sep = ""),
         x = "Rejection Sampling", y = "SIR")

ggplot() + theme_test() + labs(x = "lambda") +
  theme(plot.title = element_text(hjust = 0.5), plot.subtitle = element_text(hjust = 0.5),
        legend.position = "top") +
  geom_density(aes(l.sir, fill = "SIR"), size = 1, alpha = 0.7) +
  geom_density(aes(l.rs, fill = "Rej.Sampling"), size = 1, alpha = 0.3) +
  scale_fill_brewer("", palette = "Blues")


n10000 = myfunc1(m, 10000) ; n1000 = myfunc1(m, 1000) ; n50000 = myfunc1(m, 50000)
ggplot() + theme_minimal() + scale_color_brewer("n", palette = "Blues") +
  geom_density(aes(n1000, color = "1000"), size = 3) +
  geom_density(aes(n10000, color = "10000"), size = 2) +
  geom_density(aes(n50000, color = "50000"), size = 1) +
  labs(x = "lambda", title = "Density Plot") +
  theme(plot.title = element_text(hjust = 0.5), legend.position = "top")
```

 # 3. Network Failure Prob.

```
myroute <- function(x) {
  m <- matrix(c(1, x[1], x[2], x[3], x[4], 0, 0, 0, 0, 0,
                x[1], 1, x[5], 0, 0, x[8], x[9], 0, 0, 0,
                x[2], x[5], 1, x[6], 0, 0, x[10], 0, 0, 0,
                x[3], 0, x[6], 1, x[7], 0, 0, x[11], 0, 0,
                x[4], 0, 0, x[7], 1, 0, 0, x[12], x[13], 0,
                0, x[8], 0, 0, 0, 1, x[14], 0, 0, x[17],
                0, x[9], x[10], 0, 0, x[14], 1, x[15], 0, x[18],
                0, 0, 0, x[11], x[12], 0, x[15], 1, x[16], x[19],
                0, 0, 0, 0, x[13], 0, 0, x[16], 1, x[20],
                0, 0, 0, 0, 0, x[17], x[18], x[19], x[20], 1),
              ncol = 10, byrow = T)
  res <- ifelse((m %*% m %*% m %*% m %*% m %*% m %*% m %*% m %*% m)[1, 10] > 0, 0, 1)
  return(res)
}

n = 100000 ; p = 0.05 ; p.star = 0.2

x.mc <- matrix(sample(c(0, 1), 20*n, replace = T, prob = c(p, 1-p)), ncol = n) # 0: broken
hx.mc <- apply(x.mc, myroute, MARGIN = 2)

x.is <- matrix(sample(c(0, 1), 20*n, replace = T, prob = c(p.star, 1-p.star)), ncol = n)
hx.is <- apply(x.is, myroute, MARGIN = 2)
bx.is <- 20 - apply(x.is, sum, MARGIN = 2)
usiw <- ((1-p)/(1-p.star))^20 * ((p*(1-p.star))/(p.star*(1-p)))^bx.is
siw <- usiw / sum(usiw)

sum(hx.mc) ; sum(hx.is) # of broken edges

mu.mc <- mean(hx.mc)
mu.is.star <- mean(hx.is*usiw)
mu.is <- sum(hx.is*siw)
c(mu.mc, mu.is.star, mu.is) %>% round(8)

se.mc <- sqrt( var(hx.mc) / n )
se.is.star <- sqrt( var(hx.is*usiw) / n )
```

```
se.is <- sqrt( (var(hx.is*usiw) + mu.is^2*var(usiw) - 2*mu.is*cov(hx.is*usiw, usiw)) / n )
c(se.mc, se.is.star, se.is) %>% round(8)

myplot <- function(x) {
  xpt = c(0, 1, 1, 1, 1, 2, 2, 2, 2, 3) ; ypt = c(0, 1.5, 0.5, -0.5, -1.5, 1.5, 0.5, -0.5, -1.5, 0)
  g <- ggplot() + theme_minimal() + labs(x = "", y = "") +
       theme(axis.title = element_blank(), axis.text = element_blank(), axis.ticks = element_blank()) +
       geom_point(data = data.frame(xpt, ypt), aes(xpt, ypt), size = 5)

  if (x[1]==1) { g <- g + geom_line(aes(c(0, 1), c(0, 1.5))) }
  if (x[2]==1) { g <- g + geom_line(aes(c(0, 1), c(0, 0.5))) }
  if (x[3]==1) { g <- g + geom_line(aes(c(0, 1), c(0, -0.5))) }
  if (x[4]==1) { g <- g + geom_line(aes(c(0, 1), c(0, -1.5))) }
  if (x[5]==1) { g <- g + geom_line(aes(c(1, 1), c(1.5, 0.5))) }
  if (x[6]==1) { g <- g + geom_line(aes(c(1, 1), c(0.5, -0.5))) }
  if (x[7]==1) { g <- g + geom_line(aes(c(1, 1), c(-0.5, -1.5))) }
  if (x[8]==1) { g <- g + geom_line(aes(c(1, 2), c(1.5, 1.5))) }
  if (x[9]==1) { g <- g + geom_line(aes(c(1, 2), c(1.5, 0.5))) }
  if (x[10]==1) { g <- g + geom_line(aes(c(1, 2), c(0.5, 0.5))) }
  if (x[11]==1) { g <- g + geom_line(aes(c(1, 2), c(-0.5, -0.5))) }
  if (x[12]==1) { g <- g + geom_line(aes(c(1, 2), c(-1.5, -0.5))) }
  if (x[13]==1) { g <- g + geom_line(aes(c(1, 2), c(-1.5, -1.5))) }
  if (x[14]==1) { g <- g + geom_line(aes(c(2, 2), c(1.5, 0.5))) }
  if (x[15]==1) { g <- g + geom_line(aes(c(2, 2), c(0.5, -0.5))) }
  if (x[16]==1) { g <- g + geom_line(aes(c(2, 2), c(-0.5, -1.5))) }
  if (x[17]==1) { g <- g + geom_line(aes(c(2, 3), c(1.5, 0))) }
  if (x[18]==1) { g <- g + geom_line(aes(c(2, 3), c(0.5, 0))) }
  if (x[19]==1) { g <- g + geom_line(aes(c(2, 3), c(-0.5, 0))) }
  if (x[20]==1) { g <- g + geom_line(aes(c(2, 3), c(-1.5, 0))) }

  g <- g + geom_point(aes(c(0, 3), c(0, 0)), color = "orange", size = 5)
  return(g)
}

myplot(rep(1, 20))

# change p.star
p.star <- 0.4 ; x.is.new <- matrix(sample(c(0, 1), 20*n, replace = T, prob = c(p.star, 1-p.star)), ncol = n)
hx.is.new <- apply(x.is.new, myroute, MARGIN = 2)
bx.is.new <- 20 - apply(x.is.new, sum, MARGIN = 2)
usiw <- ((1-p)/(1-p.star))^20 * ((p*(1-p.star))/(p.star*(1-p)))^bx.is.new
siw <- usiw / sum(usiw)

mu.is.star.new <- sum(hx.is.new*usiw) / n
se.is.star.new <- sqrt( var(hx.is.new*usiw) / n )
```