Statistical Computing with R – MATH 6382[1,*] Set 5 (Monte Carlo Integration and Variance Reduction)

Tamer Oraby
UTRGV
tamer.oraby@utrgv.edu

[1] Based on textbook.

[*] Last updated November 2, 2016

Numerical Integration

Integration

- The goal is to find the integral $\int_a^b g(x)dx$ with possibly unbounded limits
- We know that $\int_a^b 2xdx = b^2 - a^2$, but we don't know a closed form for $\int_a^b e^{-x^2}dx$

1. Newton-Cotes Integration

If $a, b < \infty$. The Newton-Cotes Integration of degree n

$$\int_a^b g(x)dx \approx \sum_{j=0}^{n} w_j g(x_j)$$

where the nodal points $x_j = a + j * h$, $h = \dfrac{b-a}{n}$ and the weights are the solution of

$$\begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ x_0 & x_1 & x_2 & \cdots & x_n \\ x_0^2 & x_1^2 & x_2^2 & \cdots & x_n^2 \\ & & \vdots & & \\ x_0^n & x_1^n & x_2^n & \cdots & x_n^n \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix} = (b-a) \begin{pmatrix} 1 \\ \frac{1}{2} \\ \frac{1}{3} \\ \vdots \\ \frac{1}{n+1} \end{pmatrix}$$

1. Newton-Cotes Integration

If $a, b < \infty$. The Newton-Cotes Integration of degree n

$$\int_a^b g(x)dx \approx \sum_{j=0}^{n} w_j g(x_j)$$

where the nodal points $x_j = a + j * h$, $h = \dfrac{b-a}{n}$ and the weights are the solution of

$$w_j = \int_a^b L_j(x)dx$$

where $L_j(x) = \prod_{k=0,k\neq j}^{n} \dfrac{(x - x_k)}{(x_j - x_k)}$

1. Newton-Cotes Integration

- The **error** in the Newton-Cotes Integration of a function $g \in C^{n+1}[a, b]$ of degree n is

$$\int_a^b g(x)dx - \sum_{j=0}^{n} w_j g(x_j) =$$

$$\frac{h^{n+2} g^{(n+1)}(\xi)}{n^{n+2}(n+1)!} \int_0^n t(t-1)\cdots(t-n)dt$$

  if n is odd, where $\xi \in (a, b)$.

- Thus, Newton-Cotes has a precision of n if n is odd as it integrates polynomials of degrees up to n exactly.

1. Newton-Cotes Integration

- The **error** in the Newton-Cotes Integration of a function $g \in C^{n+2}[a,b]$ of degree n is

$$\int_a^b g(x)dx - \sum_{j=0}^{n} w_j g(x_j) =$$

$$\frac{h^{n+3} g^{(n+2)}(\xi)}{n^{n+3}(n+2)!} \int_0^n t^2(t-1)\cdots(t-n)dt$$

  if n is even, where $\xi \in (a,b)$.
- Thus, Newton-Cotes has a precision of $n+1$ if n is even as it integrates polynomials of degrees up to $n+1$ exactly.

2. Gaussian Quadrature Integration

The Gaussian Quadrature (Gauss-"Polynomial") Integration of degree n

$$\int_a^b g(x)\rho(x)dx \approx \sum_{j=0}^{n} w_j g(x_j)$$

where the nodal points $x_0, x_1, \ldots, x_n$ are the **zeros of an orthogonal polynomial** $p_{n+1}(x)$ with weight function $\rho(x)$; that is,

$$\int_a^b p_i(x)p_j(x)\rho(x)dx = 0 \text{ for } i \neq j$$

and the weights are

$$w_j = \int_a^b L_j(x)\rho(x)dx$$

where $L_j(x) = \prod_{k=0,k\neq j}^{n} \dfrac{(x - x_k)}{(x_j - x_k)}$

2. Gaussian Quadrature Integration

The Gaussian Quadrature (Gauss-"Polynomial") Integration

$$\int_a^b g(x)\rho(x)dx$$

Gaussian Quad.	**Polynomial**	$\rho(\mathbf{x})$	$[a, b]$
Gauss-Legendre	Legendre	1	$[-1, 1]$
Gauss-Chebyshev	Chebyshev (first kind)	$\dfrac{1}{\sqrt{1-x^2}}$	$(-1, 1)$
Gauss-Chebyshev	Chebyshev (second kind)	$\sqrt{1-x^2}$	$[-1, 1]$
Gauss-Laguerre	Laguerre	e^{-x}	$[0, \infty)$
Gauss-Hermite	Hermite	e^{-x^2}	$(-\infty, \infty)$

2. Gaussian Quadrature Integration

- The **error** in the Gaussian Quadrature Integration of a function $g \in C^{2n+2}[a,b]$ of degree n is

$$\int_a^b g(x)\rho(x)dx - \sum_{j=0}^{n} w_j g(x_j) = \frac{g^{(2n+2)}(\xi)}{(2n+2)!}\int_a^b p_{n+1}^2(x)\rho(x)dx$$

  where $\xi \in (a,b)$.
- Thus, Gauss-Quadrature has a precision of $2n+1$ as it integrates polynomials of degrees up to $2n+1$ exactly.
- For Legendre polynomial, $\int_{-1}^{1} p_{n+1}^2(x)\rho(x)dx = \frac{2}{2n+3}$.
- For Hermite polynomial, $\int_{-\infty}^{\infty} p_{n+1}^2(x)\rho(x)dx = \sqrt{\pi}2^{n+1}(n+1)!$.

2. Gaussian Quadrature Integration

To find

$$\int_1^3 e^{-x^2} dx$$

use either

1. Gauss-Legendre with $\rho(x) = 1$ and $g(x) = e^{-(x+2)^2}$ as

$$\int_1^3 e^{-x^2} dx = \int_{-1}^1 e^{-(x+2)^2} * 1 dx$$

or

2. Gauss-Hermite with $\rho(x) = e^{-x^2}$ and $g(x) = I_{(1,3)}(x)$ as

$$\int_1^3 e^{-x^2} dx = \int_{-\infty}^{\infty} I_{(1,3)}(x) * e^{-x^2} dx$$

Adaptive Quadrature Integration in R

To find $\int_a^b g(x)dx$ use

```
integrate(function(x) g(x),a,b)
```

Example: Find $\int_0^1 2xdx$

```
> integrate(function(x) 2*x,0,1)
1 with absolute error < 1.1e-14
```

Monte Carlo Integration

Monte Carlo Integration

To find $\int_0^1 g(x)dx$ we can estimate just $\mathbf{E}(g(X)) = \int_0^1 g(x)dx$ with $X \sim unif(0,1)$ by generating $x_1, \ldots, x_n$ from $unif(0,1)$ and then use $\frac{1}{n}\sum_{i=1}^n g(x_i)$ as estimate of $\mathbf{E}(g(X))$.

Recall (SLLN): (keep in mind $Y = g(X)$)
If $Y, Y_1, Y_2, \ldots$ are i.i.d.r.v.'s such that $\mathbf{E}\,|Y| < \infty$, then

$$\frac{1}{n}\sum_{i=1}^{n} Y_i \to \mathbf{E}(Y)$$

almost surely (with probability one).

Monte Carlo Integration

Remark:
$\int_a^b g(x)dx = (b-a)\int_0^1 g(y(b-a)+a)dy \approx (b-a)\frac{1}{n}\sum_{i=1}^n g(u_i(b-a)+a)$
with $u_1, \ldots, u_n$ generated from $unif(0,1)$

OR

$\int_a^b g(x)dx = (b-a)\int_a^b g(x)\frac{1}{b-a}dx \approx (b-a)\frac{1}{n}\sum_{i=1}^n g(x_i)$ with $x_1, \ldots, x_n$ generated from $unif(a,b)$

Monte Carlo Integration

Recall (CLT): (keep in mind $Y = g(X)$)
If $Y, Y_1, Y_2, \ldots$ are i.i.d.r.v.'s such that $\mathbf{V}(Y) < \infty$, then

$$\frac{\frac{1}{n}\sum_{i=1}^{n} Y_i - \mathbf{E}(Y)}{\sqrt{\mathbf{V}(Y)/n}} \to Z \text{ in distribution}$$

where $Z \sim norm(0, 1)$.

Note also, that $\mathbf{E}(\frac{1}{n}\sum_{i=1}^{n} Y_i) = \mathbf{E}_X(g(X))$ so it is unbiased estimate.

Monte Carlo Integration

It is good to build up a confidence interval for $\mathbf{E}(g(X))$ with $\sqrt{\mathbf{V}(Y)/n}$ estimated by the standard error

$$\hat{se}(\bar{Y}) = \sqrt{\frac{\hat{\mathbf{V}}(Y)}{n}} = \frac{1}{\sqrt{n}}\sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \bar{Y})^2}$$

for sufficiently large n.
So, $(1-\alpha)100\%$ confidence interval is found by

$$\frac{1}{n}\sum_{i=1}^{n} Y_i \pm z_{\alpha/2}\hat{se}(\bar{Y})$$

Monte Carlo Integration

Example: Estimate $\int_1^3 e^{-x^2} dx$ and find 95% confidence interval for that integral. Remark: $\int_1^3 e^{-x^2} dx = \sqrt{\pi} P(1 < X < 3)$ when $X \sim N(0, \frac{1}{\sqrt{2}})$ and

```
sqrt(pi)*(pnorm(3,0,1/sqrt(2))-pnorm(1,0,1/sqrt(2)))
 [1] 0.1393832
```

Monte Carlo Integration

First method:

$$\int_1^3 e^{-x^2}dx = \int_1^3 (3-1)*e^{-x^2}\frac{1}{3-1}dx = \mathbf{E}_X((3-1)*e^{-X^2})$$

with $X \sim unif(1,3)$

```
n<-10000;CL<-.95
x<-runif(n,1,3)
y<-(3-1)*exp(-1*x^2)
mu1<-mean(y)
mu1
 [1] 0.1363614
se1<-sd(y)/sqrt(n)
CI<-c(mu1-qnorm((1+CL)/2)*se1,mu1+qnorm((1+CL)/2)*se1)
CI
 [1] 0.1326229 0.1401000
```

Monte Carlo Integration

Second method:

$$\int_1^3 e^{-x^2} dx = \int_{-\infty}^{\infty} (\sqrt{\pi} I_{(1,3)}(x)) \frac{1}{\sqrt{\pi}} e^{-x^2} dx = \mathbf{E}_X(\sqrt{\pi} I_{(1,3)}(X))$$

with $X \sim norm(0, \frac{1}{\sqrt{2}})$

```
n<-10000;CL<-.95
x<-rnorm(n,0,1/sqrt(2))
y<-sqrt(pi)*as.integer((x<3)&(x>1))
mu2<-mean(y)
mu2
 [1] 0.1389604
se2<-sd(y)/sqrt(n)
CI<-c(mu2-qnorm((1+CL)/2)*se2,mu2+qnorm((1+CL)/2)*se2)
CI
 [1] 0.1296219 0.1482988
```

Variance Reduction

Efficiency

If $\hat{\theta}_1$ and $\hat{\theta}_2$ are two estimators of a parameter θ then $\hat{\theta}_1$ is more efficient than $\hat{\theta}_2$ if

$$\mathbf{V}(\hat{\theta}_1) < \mathbf{V}(\hat{\theta}_2)$$

and the amount of reduction in variance is measured by

$$(\mathbf{V}(\hat{\theta}_1) - \mathbf{V}(\hat{\theta}_2))/\mathbf{V}(\hat{\theta}_1)$$

Note that computational efficiency is also implied.

Example: In the previous example $\hat{\theta}_1 = 0.1363614$ and $\hat{\theta}_2 = 0.1389604$ are two estimators of $\int_1^3 e^{-x^2} dx$
The estimated amount of reduction in variance is

```
(se1^2-se2^2)/se1^2
[1] -5.31999
```

Variance Reduction–Antithetic Variables

Variance Reduction–Antithetic Variables

- If X and Y are negatively correlated ($Cov(X, Y) < 0$) then

$$\mathbf{V}(X+Y) = \mathbf{V}(X) + \mathbf{V}(Y) + 2Cov(X, Y) < \mathbf{V}(X) + \mathbf{V}(Y)$$

- If $U \sim unif(0, 1)$ then $1 - U \sim unif(0, 1)$ and

$$Cov(U, 1-U) = -\mathbf{V}(U) = -\frac{1}{12} < 0$$

- What about $Cov\left(F_X^{-1}(U), F_X^{-1}(1-U)\right)$?
- What about $Cov\left(g(F_X^{-1}(U)), g(F_X^{-1}(1-U))\right)$?

  If g is monotone then the last covariance is also negative.

Variance Reduction–Antithetic Variables

- What about $Cov\left(g(F_X^{-1}(U)), g(F_X^{-1}(1-U))\right)$?

  If g is monotone then the last covariance is negative. Why?
- Note that $h_1(s) = g(F_X^{-1}(s))$ and $h_2(s) = -g(F_X^{-1}(1-s))$ are monotone in a similar fashion to g.
- Note that $Y_1 = h_1(U)$ and $Y_2 = -h_2(U)$ are identically distributed
- WTS: $Cov(Y_1, Y_2) < 0$ or equivalently $\mathbf{E}(Y_1 Y_2) < \mathbf{E}(Y_1)\mathbf{E}(Y_2)$ or equivalently $\mathbf{E}(h_1(U)h_2(U)) > \mathbf{E}(h_1(U))\mathbf{E}(h_2(U))$

Variance Reduction–Antithetic Variables

- Assume WLOG that h_1 and h_2 are increasing, then for any x and $y \in \mathbb{R}$
$$(h_1(x) - h_1(y))(h_2(x) - h_2(y)) \geq 0$$
- Let U_1 and U_2 are i.i.d.r.v.'s then
$$\mathbf{E}\left((h_1(U_1) - h_1(U_2))(h_2(U_1) - h_2(U_2))\right) \geq 0$$
thus
$$\mathbf{E}\left((h_1(U_1)h_2(U_1) + h_1(U_2)h_2(U_2)\right) \geq$$
$$\mathbf{E}\left(h_1(U_2)h_2(U_1) + h_1(U_1)h_2(U_2)\right)$$
hence, by independence and identical distribution of U_1 and U_2
$$\mathbf{E}(h_1(U_1)h_2(U_1)) > \mathbf{E}(h_1(U_1))\mathbf{E}(h_2(U_1))$$

Variance Reduction–Antithetic Variables

Application: If $g(x)$ is monotone.
Using $U_1, \ldots, U_n \sim unif(0,1)$ to find $\hat{\theta}_{MC} = \frac{1}{n}\sum_{i=1}^{n} g(U_i)$ to estimate $\theta = \int_0^1 g(x)dx$, results in higher variance than using the antithetic estimator

$$\hat{\theta}_A = \frac{1}{n}\sum_{i=1}^{n/2}(g(U_i) + g(1-U_i))$$

Variance Reduction–Antithetic Variables

That is $\hat{\theta}_A$ is more efficient than $\hat{\theta}_{MC}$. Since,

$$\begin{aligned}
\mathbf{V}(\hat{\theta}_A) &= \frac{1}{n^2}\sum_{i=1}^{n/2}\mathbf{V}\left(g(U_i)+g(1-U_i)\right) \text{ by independence} \\
&= \frac{1}{2n}\mathbf{V}\left(g(U)+g(1-U)\right) \text{ since identically distributed} \\
&= \frac{1}{2n}\left[\mathbf{V}\left(g(U)\right)+\mathbf{V}\left(g(1-U)\right)+2Cov\left(g(U),g(1-U)\right)\right] \\
&\leq \frac{1}{2n}\left[\mathbf{V}\left(g(U)\right)+\mathbf{V}\left(g(1-U)\right)\right] \\
&\leq \frac{\mathbf{V}(g(U))}{n} = \mathbf{V}(\hat{\theta}_{MC})
\end{aligned}$$

Variance Reduction–Antithetic Variables

- Note that $\mathbf{V}(\hat{\theta}_A) = \frac{1}{2n}\mathbf{V}\left(g(U) + g(1-U)\right)$
- What about $Cov\left(g(F_X^{-1}(U_1), \ldots, F_X^{-1}(U_n)), g(F_X^{-1}(1-U_1), \ldots, F_X^{-1}(1-U_n))\right)$?
- If g is monotone then the last covariance is also negative. You can use induction on n.

Variance Reduction–Antithetic Variables

Example: Find the antithetic estimate of $\int_1^3 e^{-x^2} dx$.

```
n<-10000;CL<-.95
g<-function(x) (3-1)*exp(-1*x^2)
x<-runif(n/2,0,1)
u<-c(x,1-x)
u<-(3-1)*u+1
y<-g(u)
mu3<-mean(y)
mu3
 [1] 0.1384035
se3<-sqrt(var(g(x)+g(1-x))/(2*n))
CI<-c(mu3-qnorm((1+CL)/2)*se3,mu3+qnorm((1+CL)/2)*se3)
CI
 [1] 0.1368365 0.1399705
```

Variance Reduction–Antithetic Variables

The standard error of the MC

```
> se1
[1] 0.001958881
> se2
[1] 0.004887449
```

The standard error of the antithetic

```
> se3
[1] 0.0007994974
```

and reduction in variance is

```
(se3^2-se1^2)/se3^2
[1] -5.00319
```

Variance Reduction–Control Variates

Variance Reduction–Control Variates

A $\hat{\theta}_C$ estimates $\theta = \mathbf{E}(g(X))$ via a control variate $f(X)$ with a known $\mu = \mathbf{E}(f(X))$ for some function f, is given by

$$\hat{\theta}_C = \frac{1}{n}\sum_{i=1}^{n}(g(X_i) + c(f(X_i) - \mu))$$

for some c, where $X, X_1, \ldots, X_n$ are i.i.d.r.v.

- It is unbiased estimator since

$$\mathbf{E}(\hat{\theta}_C) = \frac{1}{n}\sum_{i=1}^{n}\mathbf{E}(g(X_i) + c(f(X_i) - \mu)) = \mathbf{E}(g(X)) + c\mathbf{E}((f(X) - \mu))$$
$$= \mathbf{E}(g(X))$$

- To make it efficient

$$n\mathbf{V}(\hat{\theta}_C) = \mathbf{V}(g(X) + c(f(X) - \mu)) =$$
$$\mathbf{V}(g(X)) + c^2\mathbf{V}(f(X)) + 2cCov(g(X), f(X))$$

Variance Reduction–Control Variates

The quadratic function

$$h(c) := \mathbf{V}(g(X)) + c^2\mathbf{V}(f(X)) + 2cCov(g(X), f(X))$$

attains its minimum at

$$c^* = -\frac{Cov(g(X), f(X))}{\mathbf{V}(f(X))}$$

and

$$\mathbf{V}(\hat{\theta}_{C^*}) = \underbrace{\frac{1}{n}\mathbf{V}(g(X))}_{\mathbf{V}(\hat{\theta}_{MC})} - \frac{1}{n}\frac{(Cov(g(X), f(X)))^2}{\mathbf{V}(f(X))} \leq \mathbf{V}(\hat{\theta}_{MC})$$

Variance Reduction–Control Variates

and the reduction in variance is given by

$$\frac{\mathbf{V}(\hat{\theta}_{MC}) - \mathbf{V}(\hat{\theta}_{C^*})}{\mathbf{V}(\hat{\theta}_{MC})} = \frac{(Cov(g(X), f(X)))^2}{\mathbf{V}(f(X))\mathbf{V}(g(X))}$$

$$= (Corr(g(X), f(X)))^2$$

so the higher the magnitude of correlation the higher is the reduction.

Variance Reduction–Control Variates

Example: Find estimate of $\int_1^3 e^{-x^2} dx$ using control variate.
Here, let $U \sim unif(1,3)$ and so $g(x) = 2e^{-x^2}$. Let the control variate be $f(x) = x$ as it is easy to handle since $\mu = \mathbf{E}(f(U)) = \mathbf{E}(U) = 2$ and $\mathbf{V}(f(U)) = \mathbf{V}(U) = \frac{1}{3}$. Then

$$\hat{\theta}_{C^*} = \frac{1}{n}\sum_{i=1}^{n}(g(U_i) + c^*(f(U_i) - \mu)) = \frac{1}{n}\sum_{i=1}^{n}\left(2e^{-U_i^2} + c^*(U_i - 2)\right)$$

where

$$c^* = -\frac{Cov(g(U), f(U))}{\mathbf{V}(f(U))} =$$

$$-\frac{\mathbf{E}(2Ue^{-U^2}) - \mathbf{E}(2e^{-U^2})\mathbf{E}(U)}{1/3} = 6 \underbrace{\mathbf{E}(2e^{-U^2})}_{\text{estimated by } \hat{\theta}_{C^*}} -\frac{3}{2}(e^{-1} - e^{-9})$$

Variance Reduction–Control Variates

Thus,

$$\hat{\theta}_{C*} = \frac{\frac{1}{n}\sum_{i=1}^{n}\left(2e^{-U_i^2} - \frac{3}{2}(e^{-1} - e^{-9})(U_i - 2)\right)}{1 - \frac{6}{n}\sum_{i=1}^{n}(U_i - 2)}$$

```
n<-10000;CL<-.95
x<-runif(n,1,3)
y<-mean(x)
z<-mean(2*exp(-1*x^2))
mu4<-(z-(3/2)*(exp(-1)-exp(-9))*(y-2))/(1-6*(y-2))
mu4
 [1] 0.1379537
 (cor(x,2*exp(-1*x^2)))^2
 [1] 0.7138062
```

Variance Reduction–Control Variates

```
n<-10000;CL<-.95
thetaC<-replicate(100,{
x<-runif(n,1,3)
y<-mean(x)
z<-mean(2*exp(-1*x^2))
mu4<-(z-(3/2)*(exp(-9)-exp(-1))*(y-2))/(1-6*(y-2))})
mu4<-mean(thetaC)
mu4
[1] 0.1393736
se4<-sd(thetaC)
se4
[1] 0.001039555
CI<-c(mu4-qnorm((1+CL)/2)*se4,mu4+qnorm((1+CL)/2)*se4)
CI
[1] 0.1373361 0.1414111
```

Variance Reduction–Control Variates

The standard error of the MC

```
> se1
[1] 0.001958881
> se2
[1] 0.004887449
```

The standard error of the antithetic

```
> se3
[1] 0.0007994974
```

The standard error of the control variate

```
> se4
[1] 0.001039555
```

and reduction in variance is

```
(se1^2-se4^2)/se1^2
[1] 0.7183699
```

Variance Reduction–Importance Sampling

Variance Reduction–Importance Sampling

Since

$$\theta = \int_a^b g(x)dx = \int_a^b \frac{g(x)}{f(x)} f(x)dx = \mathbf{E}_f(\frac{g(X)}{f(X)})$$

where $f(x)$ is called the importance function (a pdf) then we can estimate it with

$$\hat{\theta}_I = \frac{1}{n} \sum_{i=1}^{n} \frac{g(X_i)}{f(X_i)}$$

where $X_1, \ldots, X_n$ are generated from f.

$\hat{\theta}_I$ is an unbiased estimator of θ.

Variance Reduction–Importance Sampling

How can we choose the importance function f?

First, it must have a support coinciding with or including $[a, b]$; yet, the bigger it is, the worse it will behave.

If $[a, b] \subset [c, d]$ (the support of f) then $\int_c^d \frac{g(x)}{f(x)} I_{[a,b]}(x) f(x) dx$ will result in zeros when numbers falling outside the integration region are substituted in $I_{[a,b]}(x)$. Since that would be inefficient, then it is better to have the support of f coinciding with $[a, b]$.

Variance Reduction–Importance Sampling

Second, $\mathbf{V}(\hat{\theta}_I) = \frac{1}{n}\mathbf{V}_f(\frac{g(X)}{f(X)})$ which is the smallest possible if $\frac{g(x)}{f(x)}$ is nearly a constant as the variability in a constant is zero.

The minimum is reached at $f(x) = \dfrac{|g(x)|}{\int_a^b |g(t)|\, dt}$ that is a pdf.

Variance Reduction–Importance Sampling

Example: Find estimate of $\int_1^3 e^{-x^2} dx$ using importance sampling.

Here we will compare several importance functions including
$f_0(x) = \frac{1}{2}$, for $1 < x < 3$ (MC Integration)
$f_1(x) = e^{-x}$, for $0 < x < \infty$ (Wider domain)
$f_2(x) = 2e^{-2x}$, for $0 < x < \infty$ (Wider domain)
$f_3(x) = .5e^{-.5x}$, for $0 < x < \infty$ (Wider domain)
$f_4(x) = \frac{1}{e^{-1}-e^{-3}} e^{-x}$, for $1 < x < 3$
$f_5(x) = \frac{15}{263}(1 - x^2 + x^4/2)$, for $1 < x < 3$

Variance Reduction–Importance Sampling

```
n<-10000
g<-function(x)exp(-x^2)
x<-runif(n)
# f0
g_f<-g(2*x+1)/(1/2)
theta_0<-mean(g_f)
se_theta_0<-sd(g_f)/sqrt(n)
waste_0<-sum((g_f==0))/n
# f1
y<-1*log(1-x) # or directly rexp(n,1)
g_f<-as.integer((y>1)&(y<3))*g(y)/exp(-y)
theta_1<-mean(g_f)
se_theta_1<-sd(g_f)/sqrt(n)
waste_1<-sum((g_f==0))/n
```

Variance Reduction–Importance Sampling

```
# f2
y<-.5*log(1-x) # or directly rexp(n,2)
g_f<-as.integer((y>1)&(y<3))*g(y)/(2*exp(-2*y))
theta_2<-mean(g_f)
se_theta_2<-sd(g_f)/sqrt(n)
waste_2<-sum((g_f==0))/n
# f3
y<-2*log(1-x) # or directly rexp(n,.5)
g_f<-as.integer((y>1)&(y<3))*g(y)/(.5*exp(-.5*y))
theta_3<-mean(g_f)
se_theta_3<-sd(g_f)/sqrt(n)
waste_3<-sum((g_f==0))/n
```

Variance Reduction–Importance Sampling

```
# f4
c<-exp(-1)-exp(-3)
y<-1*log(exp(-1)-c*x)
g_f<-g(y)/((1/c)*exp(-1*y))
theta_4<-mean(g_f)
se_theta_4<-sd(g_f)/sqrt(n)
waste_4<-sum((g_f==0))/n
# f5
c<-15/263
InvF<-function(x){uniroot(function(y)
(c*(y-y^3/3+y^5/10-23/30)- x),lower=1,upper=3)$root}
xv<-as.array(x)
y<-apply(xv,1,InvF)
g_f<-g(y)/(c*(1-y^2+y^4/2))
theta_5<-mean(g_f)
se_theta_5<-sd(g_f)/sqrt(n)
```

Variance Reduction–Importance Sampling

```
waste_5<-sum((g_f==0))/n
result<-rbind(c(theta_0,theta_1,theta_2,theta_3,
theta_4,theta_5)
,c(se_theta_0,se_theta_1,se_theta_2,se_theta_3
,se_theta_4,se_theta_5)
,c(waste_0,waste_1,waste_2,waste_3,waste_4,waste_5))
result<-as.data.frame(result,row.names=c("theta"
,"se-theta","Waste"))
colnames(result)<-c("f0","f1","f2","f3","f4","f5")
result
```

```
                 f0          f1          f2          f3          f4          f5
theta    0.139417213 0.137480914 0.138070629 0.141256850 0.140172414 0.124605248
se-theta 0.001922774 0.002727444 0.003787346 0.002919893 0.001033504 0.008143972

Waste    0.000000000 0.685200000 0.868600000 0.615200000 0.000000000 0.000000000
```

Variance Reduction–Stratified Sampling

Variance Reduction–Stratified Sampling

To estimate $\theta = \int_a^b g(x) \frac{1}{b-a} dx = \mathbf{E}(g(X))$

1. Stratify (split) the interval $[a, b]$ into m sub-intervals $\ell_j = [x_{j-1}, x_j]$ with $x_j = a + j * h$ and $h = \frac{b-a}{m}$ for $j = 1, \ldots, m$.

2. Select a sub-interval I randomly and uniformly (with probability $\frac{1}{m}$), say ℓ_j, then $\mathbf{E}(g(X)) = \mathbf{E}_I(\mathbf{E}(g(X)|I)) = \frac{1}{m}\sum_{j=1}^m \mathbf{E}(g(X)|I = \ell_j)$

3. For each $j : j = 1, \ldots, m$, estimate $\mathbf{E}(g(X)|I = \ell_j)$ by $\hat{\theta}_{MC,j} = \frac{1}{n} \sum_{\{X_i \in \ell_j; i=1,\ldots,n\}} g(X_i)$ which are independent for each j (if you use different randomly generated numbers X's)

4. Estimate θ by

$$\hat{\theta}_{S,m} = \frac{1}{m} \sum_{j=1}^m \hat{\theta}_{MC,j}$$

Variance Reduction–Stratified Sampling

WTS: $\mathbf{V}(\hat{\theta}_{S,m}) < \mathbf{V}(\hat{\theta}_{MC})$

$$\begin{aligned}
\mathbf{V}(\hat{\theta}_{S,m}) &= \mathbf{V}(\frac{1}{m}\sum_{j=1}^{m}\hat{\theta}_{MC,j}) \\
&= \frac{1}{m^2}\sum_{j=1}^{m}\mathbf{V}(\hat{\theta}_{MC,j}) \text{ by independence} \\
&= \frac{1}{m^2}\sum_{j=1}^{m}\frac{\mathbf{V}(g(X)|I=\ell_j)}{n} \\
&= \frac{1}{mn}\mathbf{E}(\mathbf{V}(g(X)|I)) \\
&\le \frac{1}{mn}\mathbf{V}(g(X)) = \mathbf{V}(\hat{\theta}_{MC}) \text{ since } mn \text{ data points are used}
\end{aligned}$$

Variance Reduction–Stratified Sampling

```
n<-10000;a<-1;b<-3
g<-function(x){(b-a)*exp(-x^2)}
gx<-g(runif(n,a,b))
theta_MC<-mean(gx)
se_theta_MC<-sd(gx)/sqrt(n)
m<-4
L<-seq(a,b,length=m+1)
theta_MCJ<-c()
for (j in 1:m){
 theta_MCJ[j]<-mean(g(runif(n/m,L[j],L[j+1])))
}
theta_S<-mean(theta_MCJ)
c(theta_MC,theta_S)
[1] 0.1391709 0.1400730
```

Variance Reduction–Stratified Sampling

```
n<-10000;a<-1;b<-3;m<-4;N<-1000
g<-function(x){(b-a)*exp(-x^2)}
L<-seq(a,b,length=m+1)
Vtheta_S<-matrix(0,N,2)
for(i in 1:N){
 gx<-g(runif(n,a,b))
 Vtheta_S[i,1]<-mean(gx)
 theta_MCJ<-c()
 for (j in 1:m){
 theta_MCJ[j]<-mean(g(runif(n/m,L[j],L[j+1]))) }
 Vtheta_S[i,2]<-mean(theta_MCJ) }
apply(Vtheta_S,2,mean)
[1] 0.1393923 0.1393566
apply(Vtheta_S,2,sd)
[1] 0.0019811043 0.0008233142
```

Variance Reduction–Stratified Sampling

```
n<-10000;a<-1;b<-3;N<-1000
g<-function(x){(b-a)*exp(-x^2)}
Strat<-function(m){
L<-seq(a,b,length=m+1)
Vtheta_S<-matrix(0,N,1)
for(i in 1:N){
 theta_MCJ<-c()
 for (j in 1:m){
 theta_MCJ[j]<-mean(g(runif(n/m,L[j],L[j+1])))}
 Vtheta_S[i,1]<-mean(theta_MCJ)}
c(mean(Vtheta_S),sd(Vtheta_S))}
result<-c()
for(m in
c(2,4,8,10)){result<-c(result,c(m,Strat(m)))}
matrix(result,3,4)
```

Variance Reduction–Stratified Sampling

```
            [,1]         [,2]        [,3]         [,4]
[1,] 2.000000000 4.0000000000 8.000000000 1.000000e+01
[2,] 0.139370017 0.1393186015 0.139409651 1.393670e-01
[3,] 0.001497508 0.0007902412 0.000396519 3.229227e-04
```

End of Set 5