MATH 598: TOPICS IN STATISTICS

BASIC MONTE CARLO METHOD

To compute the integral

$$I(g) = \int g(y)f(y)dy = \mathbb{E}_f[g(Y)]$$

where $f(y)$ is a probability density, we use the Monte Carlo estimator

$$\widehat{I}_N(g) = \frac{1}{N}\sum_{i=1}^{N} g(Y_i)$$

where $Y_1, \ldots, Y_N$ are sampled (independently) from $f(.)$. Under integrability (finite expectation) conditions, we have that

$$\widehat{I}_N(g) \xrightarrow{a.s.} I(g)$$

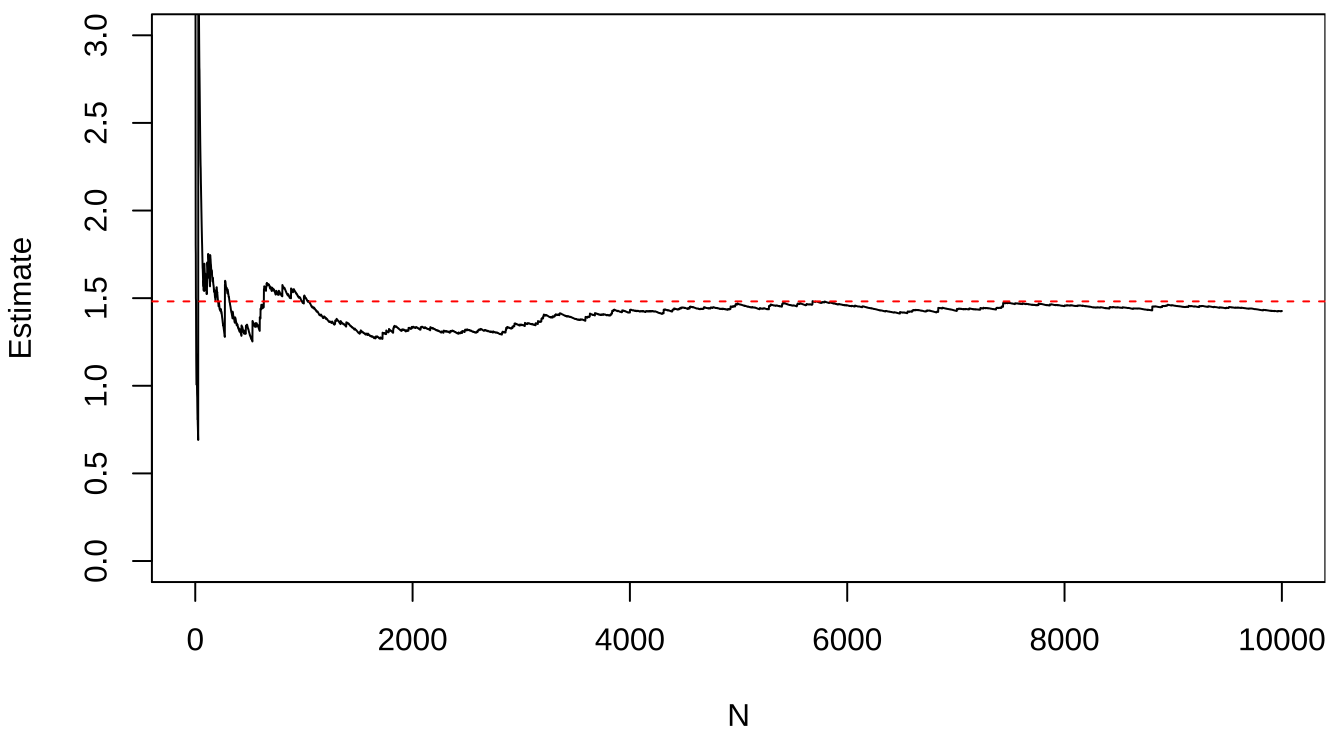
by the law of large numbers.

EXAMPLE: Suppose $Y \sim Gamma(2, 3)$. First we compute $\mathbb{E}[Y^4]$. For $r > 0$, with $Y \sim Gamma(\alpha, \beta)$

$$\mathbb{E}[Y^r] = \int_0^\infty y^r \frac{\beta^\alpha}{\Gamma(\alpha)} y^{\alpha-1} e^{-\beta y}\, dy = \frac{\beta^\alpha}{\Gamma(\alpha)} \frac{\Gamma(\alpha)}{\beta^{r+\alpha}} = \frac{1}{\beta^r}\frac{\Gamma(\alpha+r)}{\Gamma(\alpha)}$$

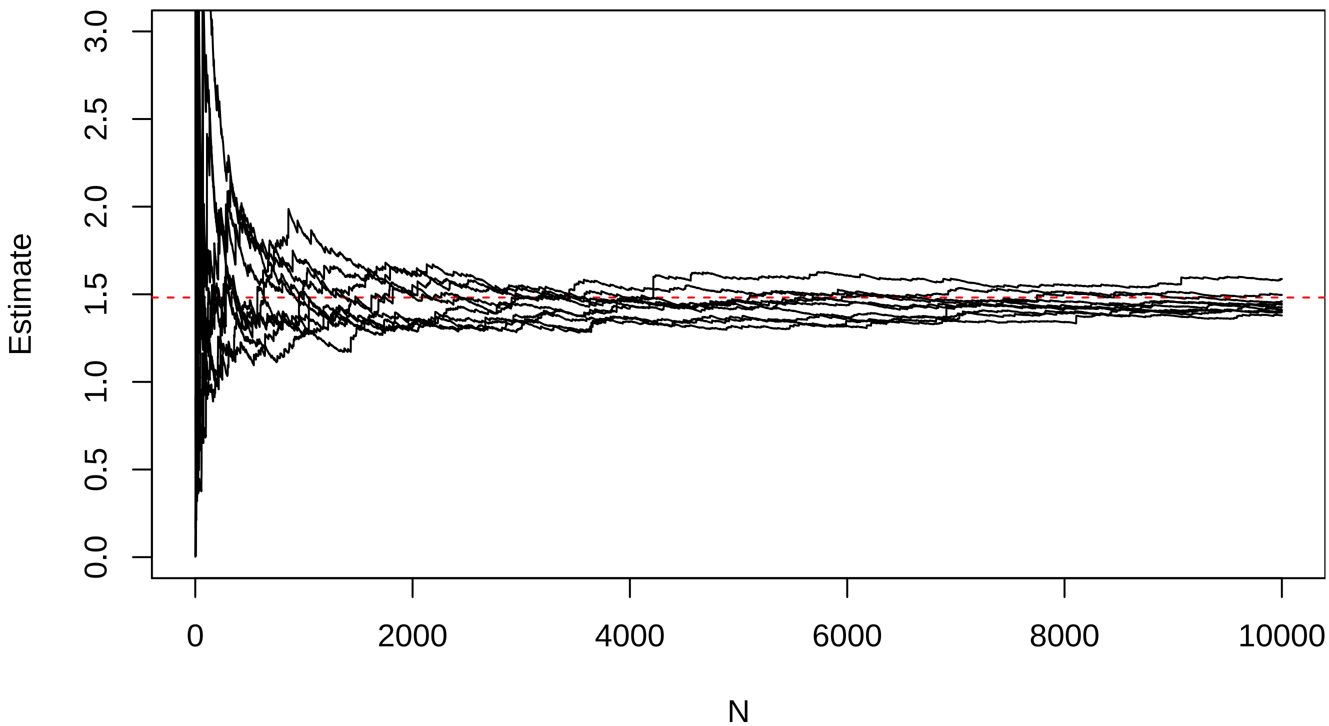
so therefore $\mathbb{E}[Y^4] = (5 \times 4 \times 3 \times 2)/3^4 = 1.481481$.

```
set.seed(64)
al<-2;be<-3;r<-4
true.val<-(gamma(al+r)/gamma(al))/be^r
N<-10000
Y<-rgamma(N,al,be)
est<-cumsum(Y^4)/c(1:N)
par(mar=c(4,4,1,0))
plot(1:N,est,type='l',xlab='N',ylab='Estimate',ylim=range(0,3))
abline(h=true.val,col='red',lty=2)
```



Some replicate runs show the variability

```r
set.seed(64)
par(mar=c(4,4,1,0))
plot(1:N,est,type='n',xlab='N',ylab='Estimate',ylim=range(0,3))
abline(h=true.val,col='red',lty=2)
for(j in 1:10){
    Y<-rgamma(N,al,be)
    tot<-cumsum(Y^4)
    est<-tot/c(1:N)
    lines(1:N,est)
    print(tot[N]/N)
}
```




```
+ [1] 1.426663
+ [1] 1.396481
+ [1] 1.496442
+ [1] 1.446813
+ [1] 1.378792
+ [1] 1.459458
+ [1] 1.412279
+ [1] 1.587252
+ [1] 1.408446
+ [1] 1.440316
```

The variance of the estimator is

$$\frac{1}{N}\text{Var}[Y^4] = \frac{1}{N}\left(\mathbb{E}[Y^8] - \{\mathbb{E}[Y^4]\}^2\right) = \frac{1}{N}\left(\frac{1}{3^8}\frac{\Gamma(10)}{\Gamma(2)} - \frac{1}{3^8}\left\{\frac{\Gamma(6)}{\Gamma(2)}\right\}^2\right) = 0.005311385$$

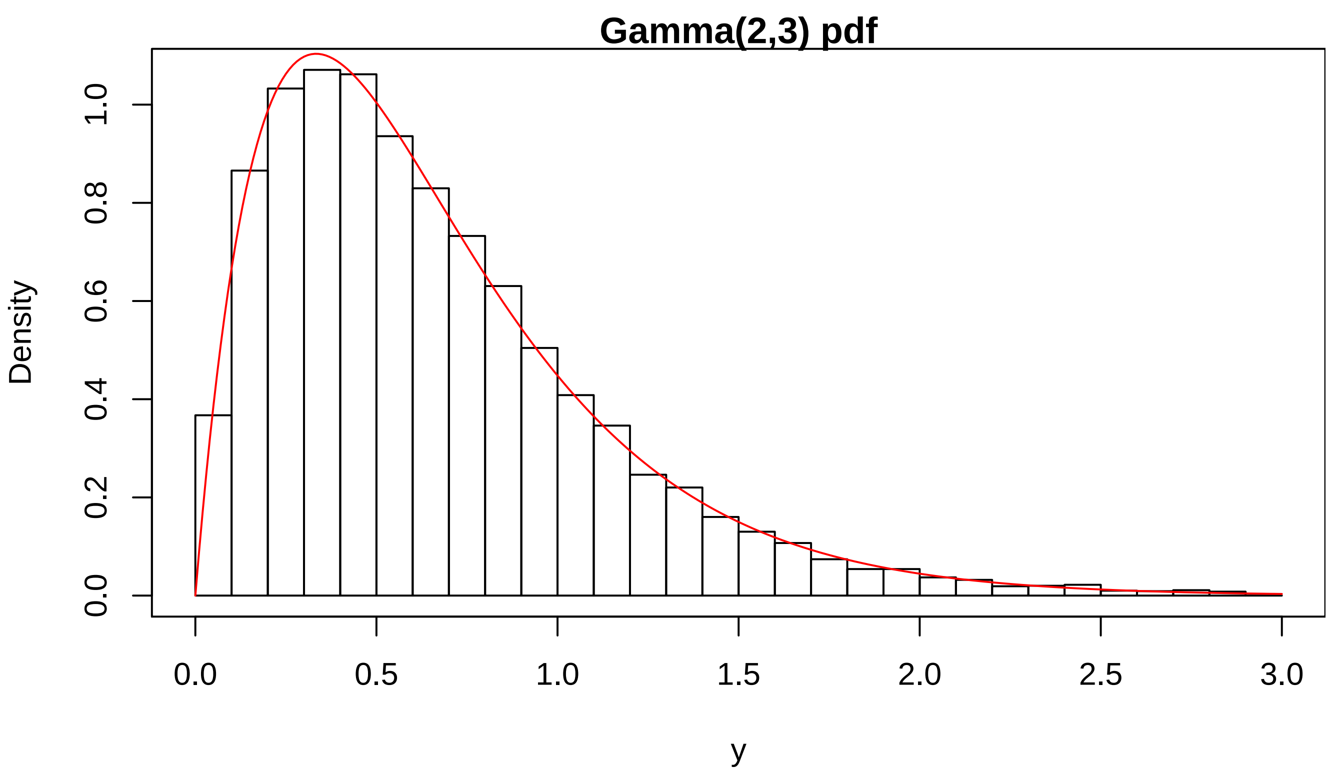
```r
(true.var<-(gamma(al+2*r)/gamma(al)-(gamma(al+r)/gamma(al))^2)/be^(2*r)/N) #True value
+ [1] 0.005311385
Ireps<-replicate(10000,mean((rgamma(N,al,be))^r));print(var(Ireps))      #Sample-based estimate
+ [1] 0.005627593
```

To compute the definite integral

$$\int_a^b f(y)\, dy$$

we can use the `pgamma` function to compute the cdf. First for $a = 0.2, b = 0.5$:

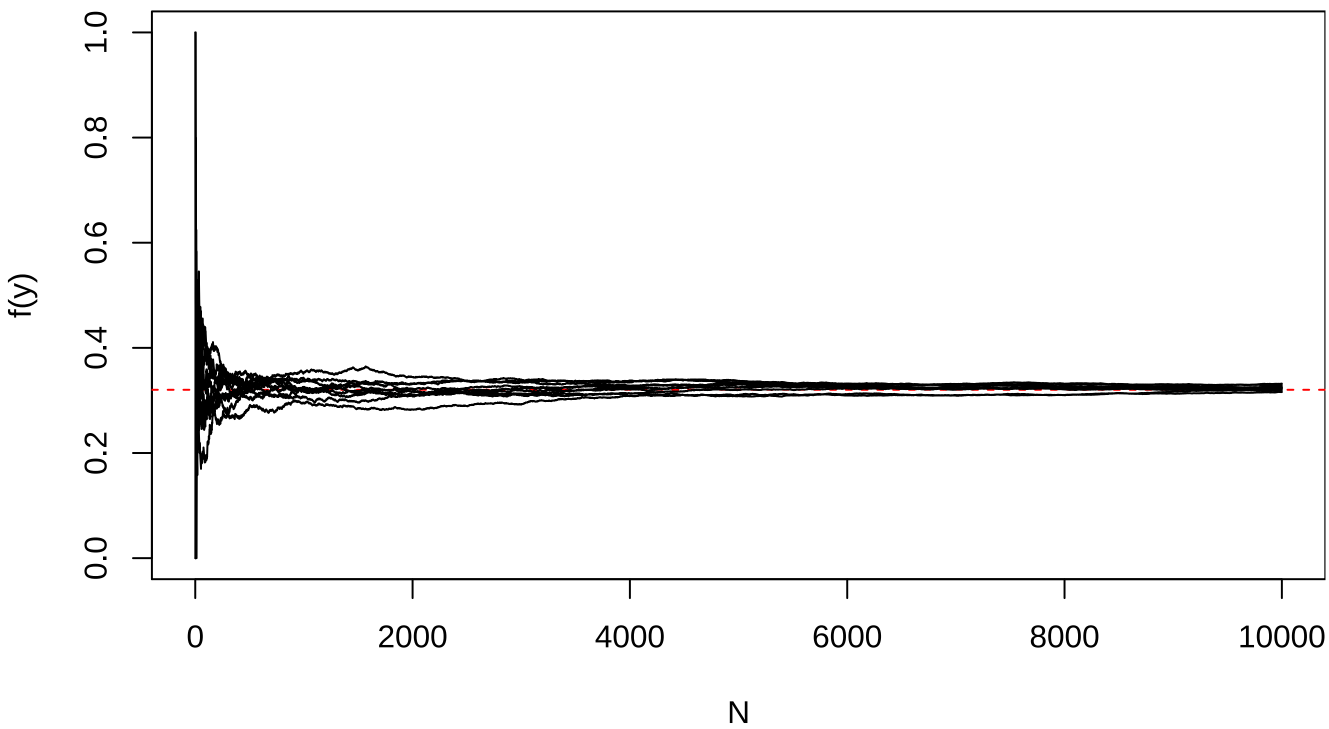
```
set.seed(64)
a<-0.2;b<-0.5
Y<-rgamma(N,al,be)
par(mar=c(4,4,1,0))
hist(Y[Y<3],breaks=seq(0,3,by=0.1),prob=T,main='Gamma(2,3) pdf',xlab='y');box()
xv<-seq(0,3,by=0.01)
yv<-dgamma(xv,al,be)
lines(xv,yv,col='red')
```




```
par(mar=c(4,4,1,0))
(true.val<-pgamma(b,al,be)-pgamma(a,al,be))

+ [1] 0.3202732

par(mar=c(4,4,1,0))
plot(1:N,est,type='n',xlab='N',ylab='f(y)',ylim=range(0,1))
abline(h=true.val,col='red',lty=2)
for(j in 1:10){
    Y<-rgamma(N,al,be)
    tot<-cumsum(Y> a & Y < b)
    est<-tot/c(1:N)
    lines(1:N,est)
    print(tot[N]/N)
}
```




```
+ [1] 0.3229
+ [1] 0.3159
+ [1] 0.3182
+ [1] 0.3243
+ [1] 0.3238
+ [1] 0.332
+ [1] 0.3212
+ [1] 0.3262
+ [1] 0.3239
+ [1] 0.3292
```

If $a = 2.5$ and $b = 3$, the estimator is much more variable relative to the estimated quantity:

```
set.seed(64)
Ind<-function(y,a,b){mean(y>a & y < b)}
a<-0.2;b<-0.5
true.val<-pgamma(b,al,be)-pgamma(a,al,be)
Ireps1<-replicate(10000,Ind(rgamma(N,al,be),a,b))
var(Ireps1)/true.val

+ [1] 0.00006807211

a<-2.5;b<-3
true.val<-pgamma(b,al,be)-pgamma(a,al,be)
Ireps2<-replicate(10000,Ind(rgamma(N,al,be),a,b))
var(Ireps2)/true.val

+ [1] 0.00009875651
```

EXAMPLE: For random variables (X, Y) that are independently uniform on the unit square, the probability

$$\Pr((X-0.5)^2 + (Y-0.5)^2 < 0.5^2) = \frac{\pi}{4} \simeq 0.785398$$

and we can compute this probability as

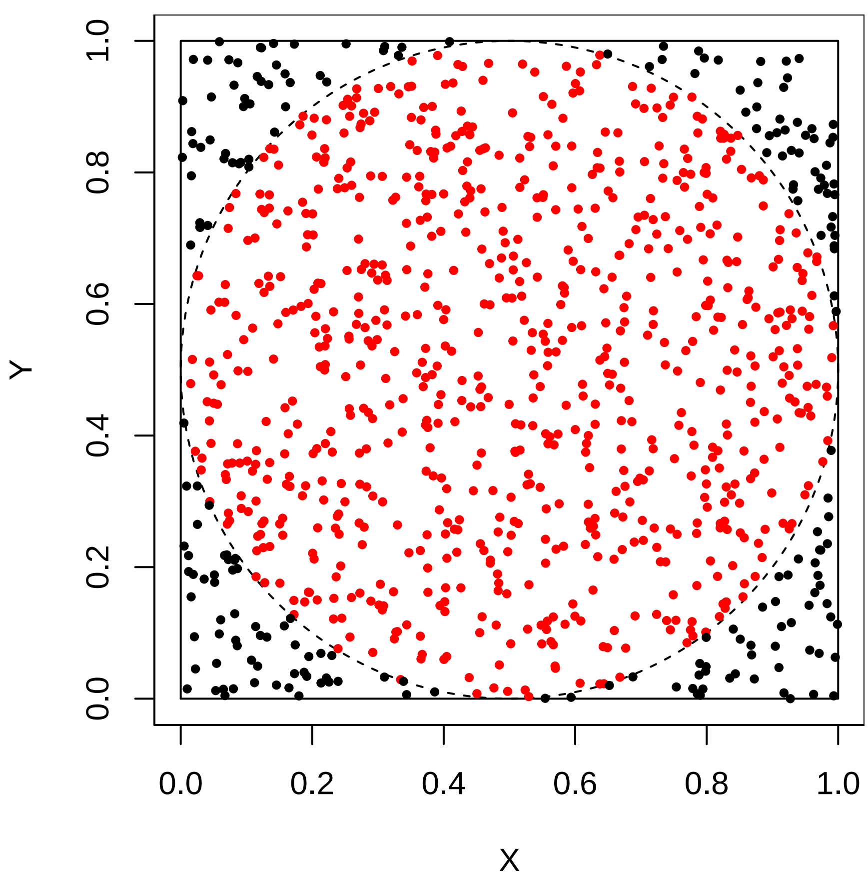
$$\iint_{\mathcal{D}} dxdy = \int_0^1 \int_0^1 \mathbb{1}_{\mathcal{D}}(x,y)\, dxdy$$

where $\mathcal{D}$ is the disk of radius 1/2 centered at $(1/2, 1/2)$. We may estimate this probability using the Monte Carlo estimator

$$\frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{\mathcal{D}}(X_i, Y_i)$$

where $(X_i, Y_i), i = 1, \ldots, N$ are drawn independently as $Uniform(0,1)$.

```r
set.seed(64)
N<-1000;X<-runif(N,0,1);Y<-runif(N,0,1)
par(mar=c(4,4,0,0),pty='s')
plot(X,Y,ylim=range(0,1),xlim=range(0,1),type='n')
lines(c(0,1,1,0,0),c(0,0,1,1,0))
tv<-seq(0,2*pi,length=1001);xv<-0.5*cos(tv)+0.5;yv<-0.5*sin(tv)+0.5
lines(xv,yv,lty=2)
inside<-(X-0.5)^2 + (Y-0.5)^2 < (0.5)^2
points(X[inside],Y[inside],col='red',pch=19,cex=0.5)
points(X[!inside],Y[!inside],col='black',pch=19,cex=0.5)
```




```r
sum(inside)/N

+ [1] 0.788
```

EXAMPLE: Consider the integral

$$\int_0^1 \frac{1}{x} \sin\left(\frac{2\pi}{x}\right)\ dx$$

Using the change of variable $t = 2\pi/x$, we see that the integral is equal to

$$\int_{2\pi}^{\infty} \frac{1}{t} \sin(t)\ dt$$

The integral

$$Si(a) = \int_0^a \frac{1}{t} \sin(t)\ dt$$

is known as the Sine Integral, a special function that can be computed numerically:

$$\int_{2\pi}^{\infty} \frac{1}{t} \sin(t)\ dt = Si(\infty) - Si(2\pi) = \frac{\pi}{2} - 1.418151 = 0.152645$$

We try to compute this integral by Monte Carlo. We write

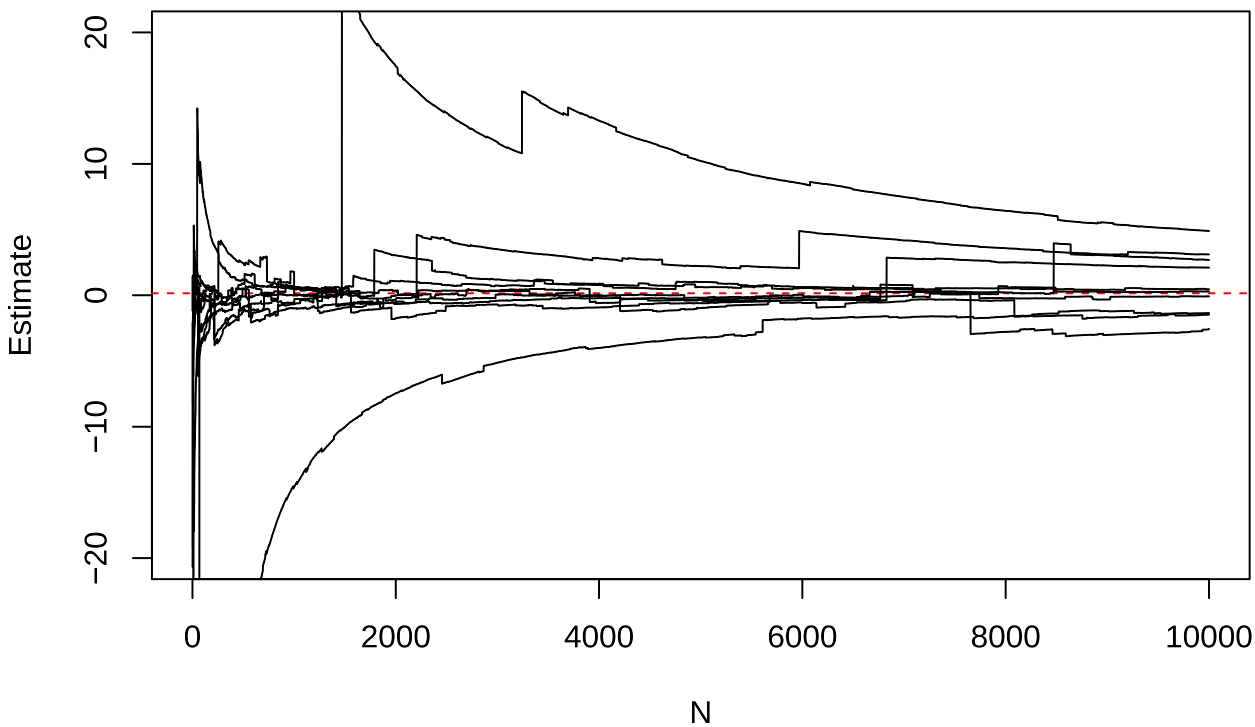
$$\int_0^1 \frac{1}{x} \sin\left(\frac{2\pi}{x}\right)\ dx = \mathbb{E}_Y\left[\frac{1}{Y} \sin\left(\frac{2\pi}{Y}\right)\right]$$

where $Y \sim Uniform(0,1)$, which implies the Monte Carlo estimator

$$\frac{1}{N}\sum_{i=1}^{N} \frac{1}{Y_i} \sin\left(\frac{2\pi}{Y_i}\right)$$

with $Y_1, \ldots, Y_N \sim Uniform(0,1)$ are independent. The simulation below shows ten multiple runs.

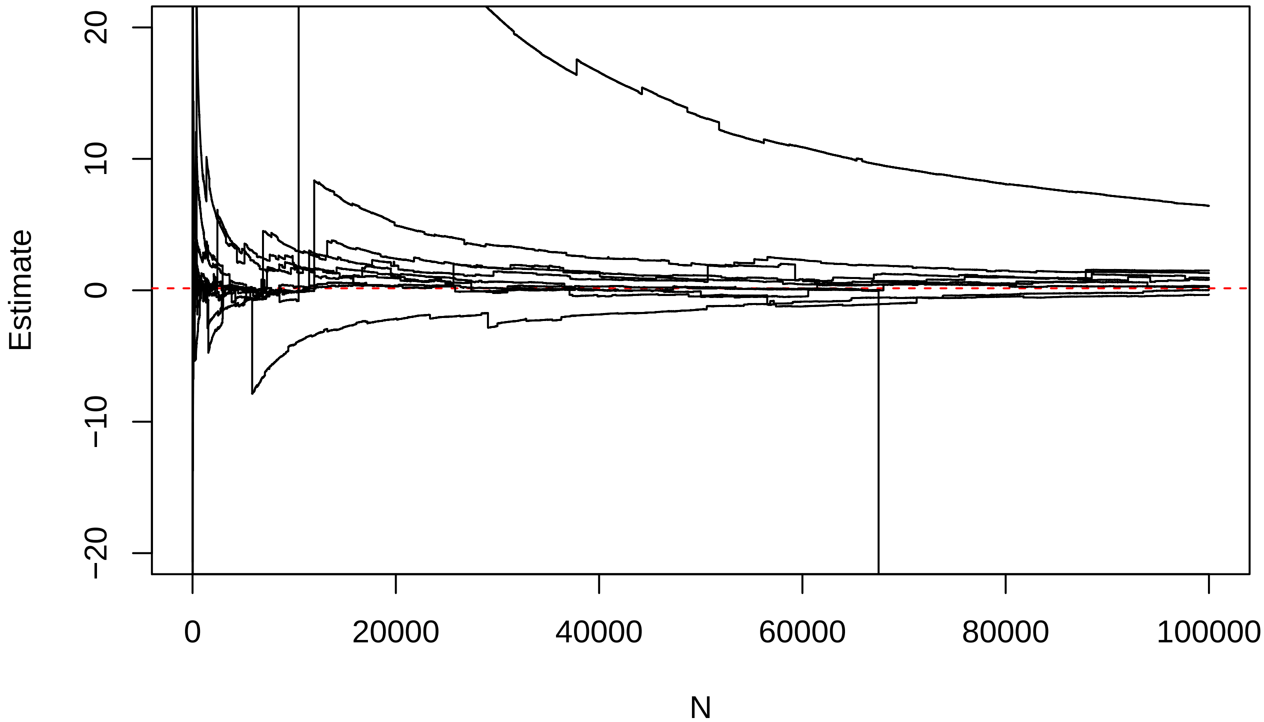
```r
set.seed(64)
par(mar=c(4,4,1,2))
N<-10000
plot(1:N,1:N,type='n',xlab='N',ylab='Estimate',ylim=range(-20,20))
true.val<-pi/2-1.418151
abline(h=true.val,col='red',lty=2)
for(j in 1:10){
    Y<-runif(N)
    gY<-sin(2*pi/Y)/Y
    tot<-cumsum(gY)
    est<-tot/c(1:N)
    lines(1:N,est)
    print(tot[N]/N)
}
```




```
+ [1] -0.08047351
+ [1] 4.89219
+ [1] 0.3007103
+ [1] 0.4608025
+ [1] -1.476808
+ [1] -1.353979
+ [1] 2.104705
+ [1] 3.108693
+ [1] 2.693405
+ [1] -2.568319
```

The Monte Carlo estimator does not seem to be performing well. This problem persists even at a sample size of $N = 100000$:

```
set.seed(64)
par(mar=c(4,4,1,2))
N<-100000
plot(1:N,1:N,type='n',xlab='N',ylab='Estimate',ylim=range(-20,20))
true.val<-pi/2-1.418151
abline(h=true.val,col='red',lty=2)
for(j in 1:10){
    Y<-runif(N)
    gY<-sin(2*pi/Y)/Y
    tot<-cumsum(gY)
    est<-tot/c(1:N)
    lines(1:N,est)
    print(tot[N]/N)
}
```




```
+ [1] 0.8080926
+ [1] 0.2490228
+ [1] 1.502258
+ [1] 6.423199
+ [1] 0.02642912
+ [1] -0.3319313
+ [1] 0.3184247
+ [1] -29.52384
+ [1] 0.9434333
+ [1] 1.310001
```

The problem remains, and is caused by the fact that the original (Riemann) integral is not *absolutely convergent*, that is,

$$\int_0^1 \left| \frac{1}{x} \sin\left(\frac{2\pi}{x}\right) \right| dx = \int_0^1 \frac{1}{x} \left| \sin\left(\frac{2\pi}{x}\right) \right| dx$$

is **not finite**. This is a necessary condition for Monte Carlo estimation (which essentially relies on Lebesgue integration) to yield an unbiased and consistent estimator. We have that if

$$\int |g(x)| f(x)\, dx$$

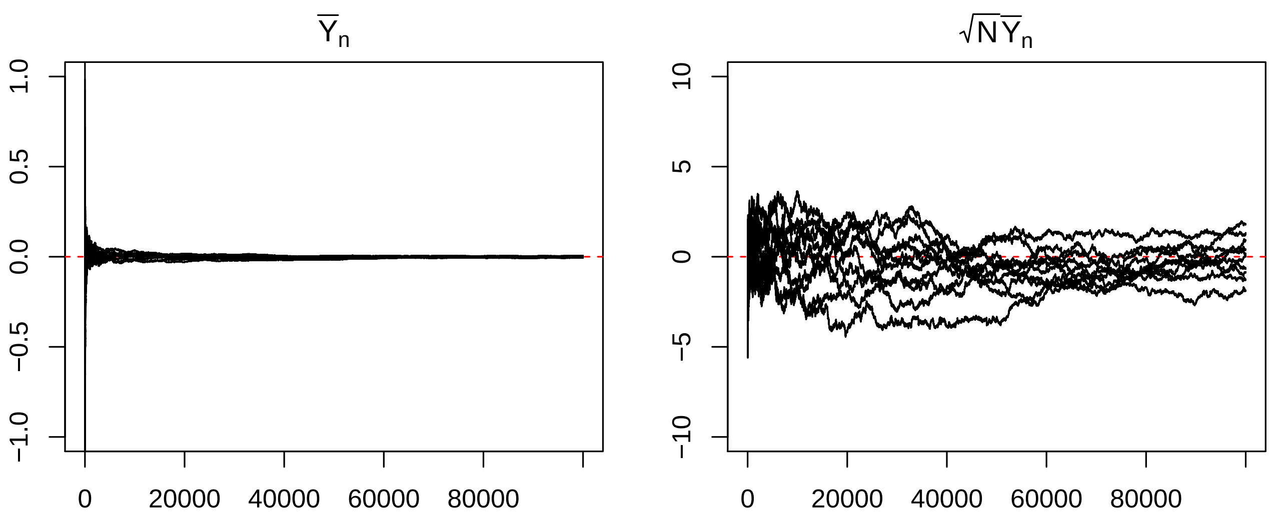
is finite, we can use Monte Carlo to estimate the integral. This coincides with the result that the expectation of a random variable Y exists and is finite if and only if the expectation of $|Y|$ is finite.

EXAMPLE: The Central Limit Theorem (CLT) states that for the Monte Carlo estimator of $I(g) = \mathbb{E}[g(Y)] < \infty$ is asymptotically Normally distributed

$$\sqrt{N}(\widehat{I}_N(g) - I(g)) \xrightarrow{d} Normal(0, V(g))$$

provided $V(g) = \text{Var}[g(Y)] < \infty$. If the variance is not finite, then the CLT approach does not apply. For example, consider estimating the mean of a standard $Student(\nu)$ distribution; the variance of this distribution is $\nu/(\nu-2)$ provided $\nu > 2$, but is infinite otherwise. When $\nu = 3$, the asymptotic variance of $\overline{Y}_n$ is 3.

```
set.seed(64)
par(mar=c(3,3,2,1),mfrow=c(1,2))
N<-100000
plot(1:N,1:N,type='n',xlab='N',ylab='Estimate',ylim=range(-1,1),main=expression(bar(Y)[n]))
abline(h=0,col='red',lty=2)
for(j in 1:10){
    Y<-rt(N,3)
    tot<-cumsum(Y)
    est<-tot/c(1:N)
    lines(1:N,est)
}
set.seed(64)
plot(1:N,1:N,type='n',xlab='N',ylab='Estimate',ylim=range(-10,10),main=expression(sqrt(N)*bar(Y)[n]))
abline(h=0,col='red',lty=2)
for(j in 1:10){
    Y<-rt(N,3)
    tot<-cumsum(Y)
    est<-tot/sqrt(c(1:N))
    lines(1:N,est)
}
```

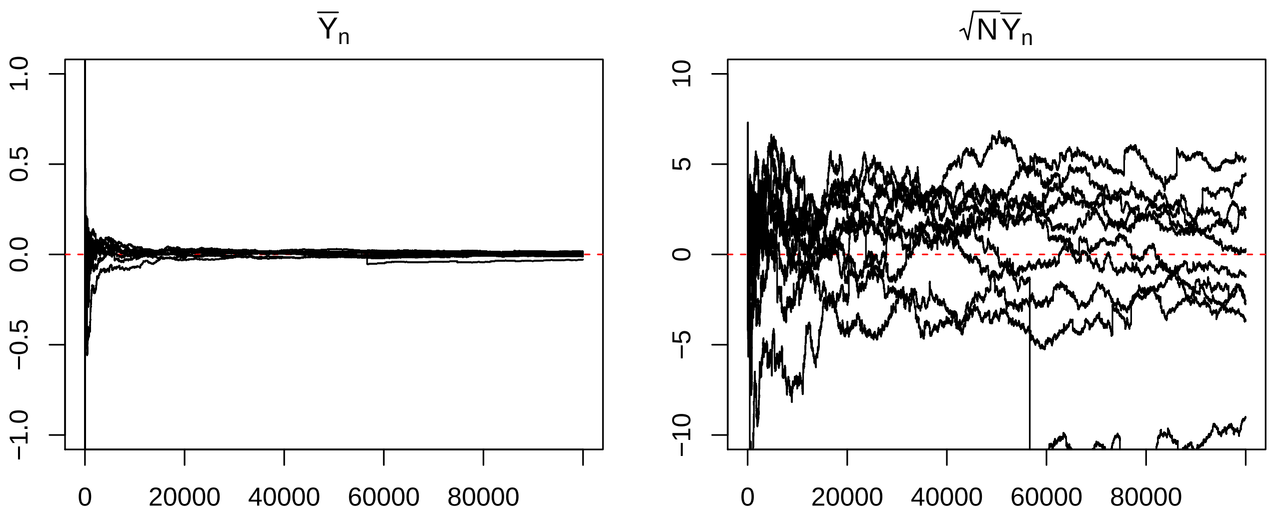



```
var(replicate(1000,sqrt(N)*mean(rt(N,3))))

+ [1] 2.977605
```

However, when $\nu = 2$, the variance of $\sqrt{N}\overline{Y}_N$ is not finite, and the standard CLT does not apply.

```
set.seed(654)
par(mar=c(3,3,2,1),mfrow=c(1,2))
plot(1:N,1:N,type='n',xlab='N',ylab='Estimate',ylim=range(-1,1),main=expression(bar(Y)[n]))
abline(h=0,col='red',lty=2)
for(j in 1:10){
    Y<-rt(N,2)
    tot<-cumsum(Y)
    est<-tot/c(1:N)
    lines(1:N,est)
}
set.seed(654)
plot(1:N,1:N,type='n',xlab='N',ylab='Estimate',ylim=range(-10,10),main=expression(sqrt(N)*bar(Y)[n]))
abline(h=0,col='red',lty=2)
for(j in 1:10){
    Y<-rt(N,2)
    tot<-cumsum(Y)
    est<-tot/sqrt(c(1:N))
    lines(1:N,est)
}
```




```
var(replicate(1000,sqrt(N)*mean(rt(N,2))))

+ [1] 27.64756

var(replicate(1000,sqrt(N)*mean(rt(N,2))))

+ [1] 18.0195
```