Topic 10

The Law of Large Numbers

10.1 Introduction

A public health official want to ascertain the mean weight of healthy newborn babies in a given region under study. If we randomly choose babies and weigh them, keeping a running average, then at the beginning we might see some larger fluctuations in our average. However, as we continue to make measurements, we expect to see this running average settle and converge to the true mean weight of newborn babies. This phenomena is informally known as the **law of averages**. In probability theory, we call this the **law of large numbers**.

Example 10.1. *We can simulate babies' weights with independent normal random variables, mean 3 kg and standard deviation 0.5 kg. The following* R *commands perform this simulation and computes a running average of the heights. The results are displayed in Figure 10.1.*

```
> n<-c(1:100)
> x<-rnorm(100,3,0.5)
> s<-cumsum(x)
> plot(s/n,xlab="n",ylim=c(2,4),type="l")
```

Here, we begin with a sequence $X_1, X_2, \ldots$ of random variables having a common distribution. Their average, the **sample mean**,

$$\bar{X} = \frac{1}{n}S_n = \frac{1}{n}(X_1 + X_2 + \cdots + X_n),$$

is itself a random variable.

If the common mean for the X_i's is μ, then by the linearity property of expectation, the mean of the average,

$$E[\frac{1}{n}S_n] = \frac{1}{n}(EX_1 + EX_2 + \cdots + EX_n) = \frac{1}{n}(\mu + \mu + \cdots + \mu) = \frac{1}{n}n\mu = \mu. \tag{10.1}$$

is also μ.

If, in addition, the X_i's are independent with common variance σ^2, then first by the quadratic identity and then the Pythagorean identity for the variance of independent random variables, we find that the variance of $\bar{X}$,

$$\sigma^2_{\bar{X}} = \text{Var}(\frac{1}{n}S_n) = \frac{1}{n^2}(\text{Var}(X_1) + \text{Var}(X_2) + \cdots + \text{Var}(X_n)) = \frac{1}{n^2}(\sigma^2 + \sigma^2 + \cdots + \sigma^2) = \frac{1}{n^2}n\sigma^2 = \frac{1}{n}\sigma^2. \tag{10.2}$$

So the mean of these running averages remains at μ but the variance is decreasing to 0 at a rate inversely proportional to the number of terms in the sum. For example, the mean of the average weight of 100 newborn babies is 3 kilograms, the standard deviation is $\sigma_{\bar{X}} = \sigma/\sqrt{n} = 0.5/\sqrt{100} = 0.05$ kilograms = 50 grams. For 10,000 males, the mean remains 3 kilograms, the standard deviation is $\sigma_{\bar{X}} = \sigma/\sqrt{n} = 0.5/\sqrt{10000} = 0.005$ kilograms = 5 grams. Notice that

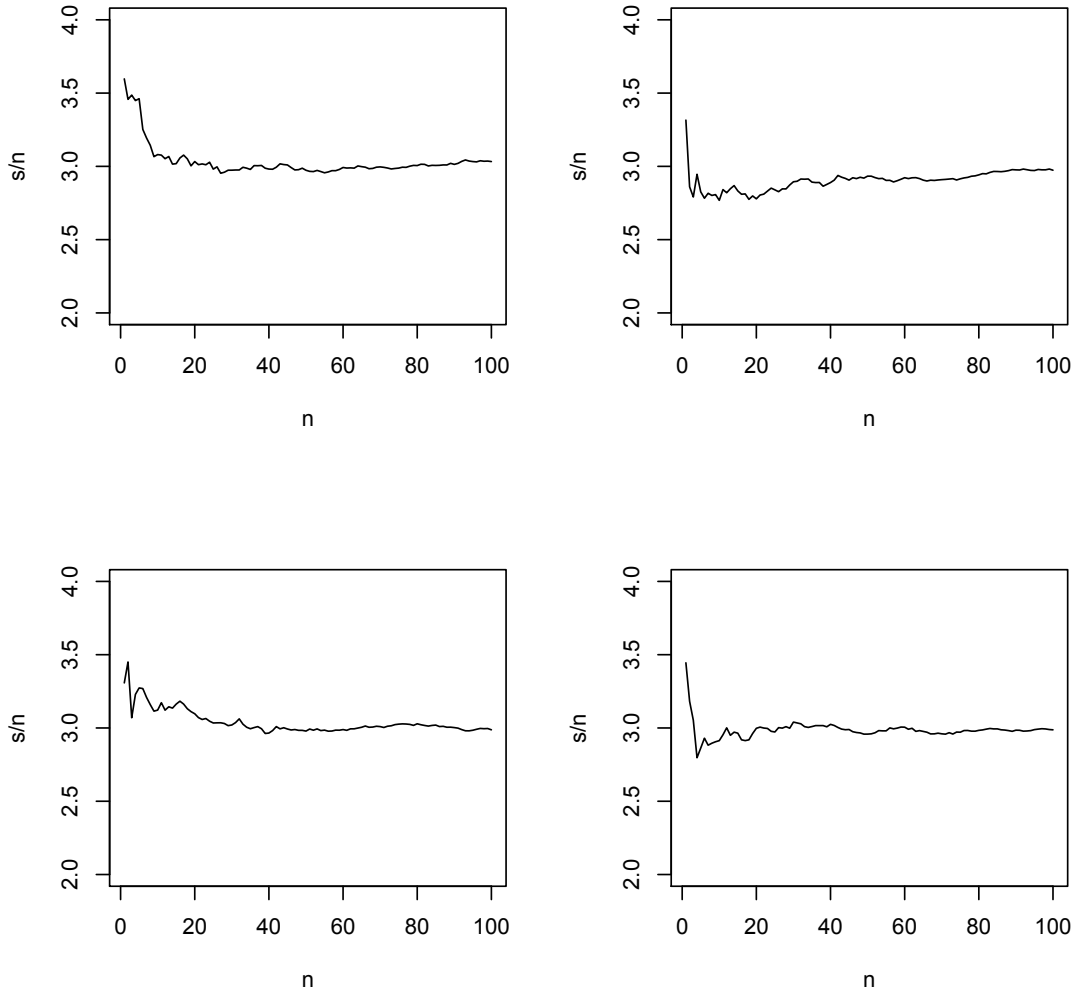

Figure 10.1: Four simulations of the running average S_n/n, $n = 1, 2, \ldots, 100$ for independent normal random variables, mean 3 kg and standard deviation 0.5 kg. Notice that the running averages have large fluctuations for small values of n but settle down converging to the mean value μ = 3 kilograms for newborn birth weight. This behavior could have been predicted using the law of large numbers. The size of the fluctuations, as measured by the standard deviation of S_n/n, is $\sigma/\sqrt{n}$ where σ is the standard deviation of newborn birthweight.

- as we increase n by a factor of **100**,
- we decrease $\sigma_{\bar{X}}$ by a factor of **10**.

The mathematical result, the **law of large numbers**, tells us that the results of these simulation could have been anticipated.

Theorem 10.2. *For a sequence of independent random variables $X_1, X_2, \ldots$ having a common distribution, their running average*

$$\frac{1}{n}S_n = \frac{1}{n}(X_1 + \cdots + X_n)$$

has a limit as $n \to \infty$ if and only if this sequence of random variables has a common mean μ. In this case the limit is μ.

The theorem also states that if the random variables do not have a mean, then as the next example shows, the limit will fail to exist. We shall show with the following example. When we look at methods for estimation, one approach, the method of moments, will be based on using the law of large numbers to estimate the mean μ or a function of μ.

Care needs to be taken to ensure that the simulated random variables indeed have a mean. For example, use the `runif` command to simulate uniform transform variables, and choose a transformation $Y = g(U)$ that results in an integral

$$\int_0^1 g(u)\, du$$

that does not converge. Then, if we simulate independent uniform random variables, the running average

$$\frac{1}{n}(g(U_1) + \cdots + g(U_n))$$

will not converge. This issue is the topic of the next exercise and example.

Exercise 10.3. *Let U be a uniform random variable on the interval $[0, 1]$. Give the value for p for which the mean is finite and the values for which it is infinite. Simulate the situation for a value of p for which the integral converges and a second value of p for which the integral does not converge and cheek has in Example 10.1 a plot of S_n/n versus n.*

Example 10.4. *The standard Cauchy random variable X has density function*

$$f_X(x) = \frac{1}{\pi}\frac{1}{1+x^2} \quad x \in \mathbb{R}.$$

Let $Y = |X|$. In an attempt to compute the improper integral for $EY = E|X|$, note that

$$\int_{-b}^{b} |x| f_X(x)\, dx = 2\int_0^b \frac{1}{\pi}\frac{x}{1+x^2}\, dx = \frac{1}{\pi}\ln(1+x^2)\Big|_0^b = \frac{1}{\pi}\ln(1+b^2) \to \infty$$

as $b \to \infty$. Thus, Y has infinite mean. We now simulate 1000 independent Cauchy random variables.

```
> n<-c(1:1000)
> y<-abs(rcauchy(1000))
> s<-cumsum(y)
> plot(s/n,xlab="n",ylim=c(-6,6),type="l")
```

These random variables do not have a finite mean. As you can see in Figure 10.2 that their running averages do not seem to be converging. Thus, if we are using a simulation strategy that depends on the law of large numbers, we need to check that the random variables have a mean.

Exercise 10.5. *Using simulations, check the failure of the law of large numbers of Cauchy random variables. In the plot of running averages, note that the shocks can jump either up or down.*

10.2 Monte Carlo Integration

Monte Carlo methods use stochastic simulations to approximate solutions to questions that are very difficult to solve analytically. This approach has seen widespread use in fields as diverse as statistical physics, astronomy, population genetics, protein chemistry, and finance. Our introduction will focus on examples having relatively rapid computations. However, many research groups routinely use Monte Carlo simulations that can take weeks of computer time to perform.

For example, let $X_1, X_2, \ldots$ be independent random variables uniformly distributed on the interval $[a, b]$ and write f_X for their common density..

Then, by the law of large numbers, for n large we have that

$$\overline{g(X)}_n = \frac{1}{n}\sum_{i=1}^{n} g(X_i) \approx Eg(X_1) = \int_a^b g(x) f_X(x)\, dx = \frac{1}{b-a}\int_a^b g(x)\, dx.$$

Thus,

$$\int_a^b g(x)\, dx. \approx (b-a)\overline{g(X)}_n.$$

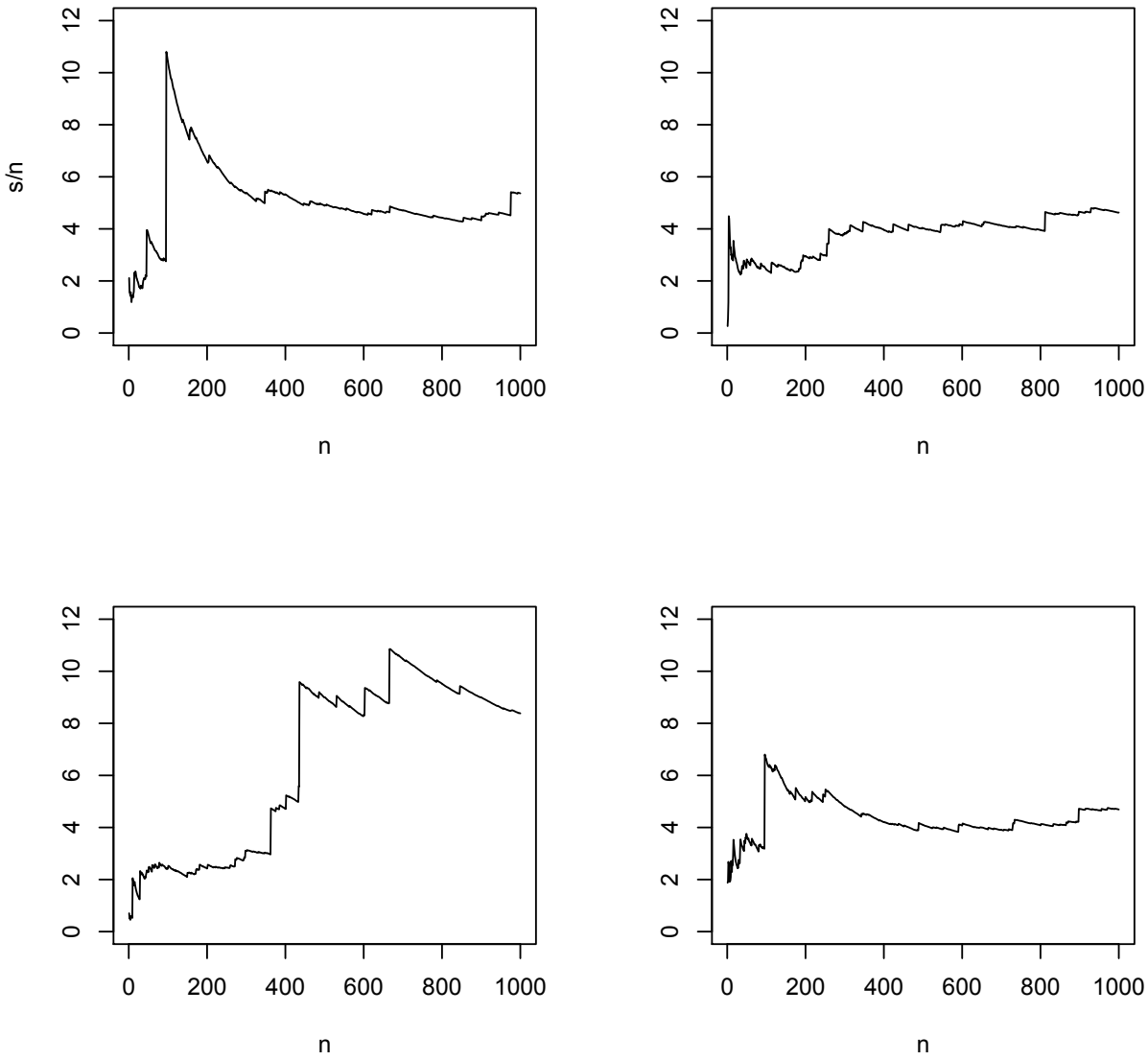

Figure 10.2: Four simulations of the running average S_n/n, $n = 1, 2, \ldots, 1000$ for the absolute value of independent Cauchy random variables. Note that the running averate does not seem to be settling down and is subject to "shocks". Because Cauchy random variables do not have a mean, we know, from the law of large numbers, that the running averages do not converge.

Recall that in calculus, we defined the average of g to be

$$\frac{1}{b-a}\int_a^b g(x)\,dx.$$

We can also interpret this integral as the expected value of $g(X_1)$.

Thus, Monte Carlo integration leads to a procedure for estimating integrals.

- Simulate uniform random variables $X_1, X_2, \ldots, X_n$ on the interval $[a, b]$.
- Evaluate $g(X_1), g(X_2), \ldots, g(X_n)$.
- Average this values and multiply by $b - a$ to estimate the integral.

Example 10.6. *Let $g(x) = \sqrt{1+\cos^3(x)}$ for $x \in [0, \pi]$, to find $\int_0^\pi g(x)\,dx$. The three steps above become the following* R *code.*

```
> x<-runif(1000,0,pi)
> g<-sqrt(1+cos(x)^3)
> pi*mean(g)
[1] 2.991057
```

Example 10.7. *To find the integral of $g(x) = \cos^2(\sqrt{x^3+1})$ on the interval $[-1, 2]$, we simulate n random variables uniformly using* `runif(n,-1,2)` *and then compute* `mean(cos(sqrt(x^3+1))^2)`. *The choices $n = 25$ and $n = 250$ are shown in Figure 10.3*

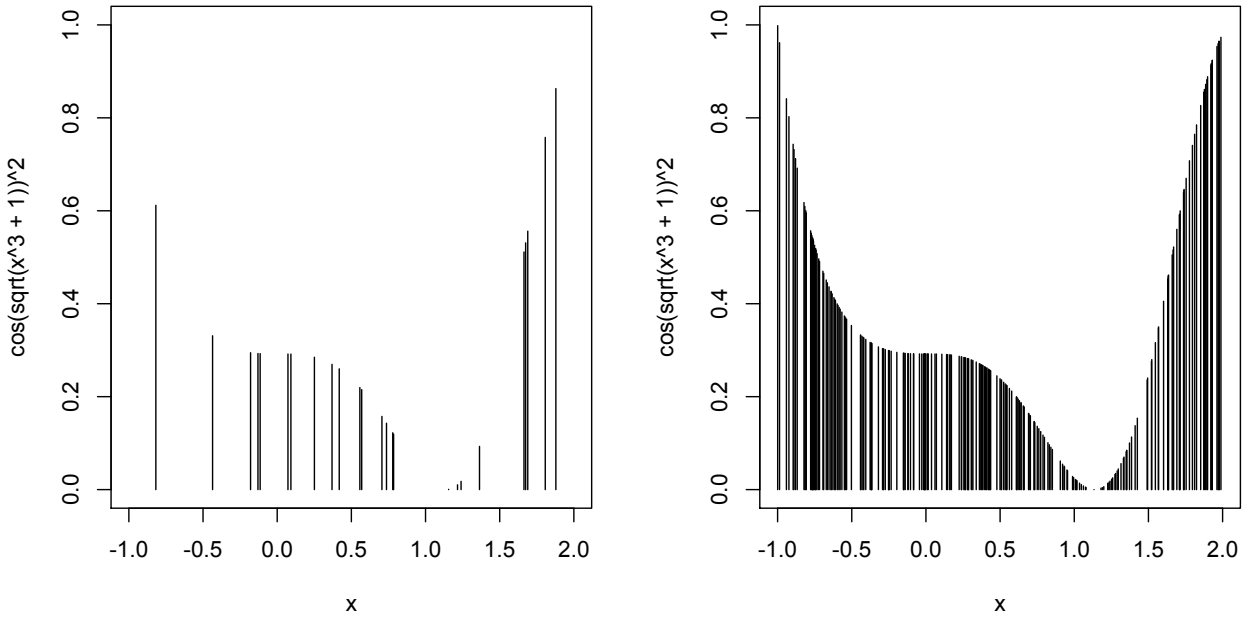

Figure 10.3: Monte Carlo integration of $g(x) = \cos^2(\sqrt{x^3+1})$ on the interval $[-1, 2]$, with (left) $n = 25$ and (right) $n = 250$. $\overline{g(X)}_n$ is the average heights of the n lines whose x values are uniformly chosen on the interval. By the law of large numbers, this estimates the average value of g. This estimate is multiplied by 3, the length of the interval to give $\int_{-1}^2 g(x)\,dx$. In this example, the estimate os the integral is 0.905 for $n = 25$ and 1.028 for $n = 250$. Using the `integrate` command, a more precise numerical estimate of the integral gives the value 1.000194.

The variation in estimates for the integral can be described by the variance as given in equation (10.2).

$$\text{Var}(\overline{g(X)}_n) = \frac{1}{n}\text{Var}(g(X_1)).$$

where $\sigma^2 = \text{Var}(g(X_1)) = E(g(X_1) - \mu_{g(X_1)})^2 = \int_a^b (g(x) - \mu_{g(X_1)})^2 f_X(x)\, dx$. Typically this integral is more difficult to estimate than $\int_a^b g(x)\, dx$, our original integral of interest. However, we can see that the variance of the estimator is inversely proportional to n, the number of random numbers in the simulation. Thus, the standard deviation is inversely proportional to $\sqrt{n}$.

Monte Carlo techniques are rarely the best strategy for estimating one or even very low dimensional integrals. R does integration numerically using the `function` and the `integrate` commands. For example,

```
> g<-function(x){sqrt(1+cos(x)^3)}
> integrate(g,0,pi)
2.949644 with absolute error < 3.8e-06
```

With only a small change in the algorithm, we can also use this to evaluate high dimensional multivariate integrals. For example, in three dimensions, the integral

$$I(g) = \int_{a_1}^{b_1} \int_{a_2}^{b_2} \int_{a_3}^{b_3} g(x, y, z)\, dz\, dy\, dx$$

can be estimated using Monte Carlo integration by generating three sequences of uniform random variables,

$$X_1, X_2, \ldots, X_n, \quad Y_1, Y_2, \ldots, Y_n, \quad \text{and} \quad Z_1, Z_2, \ldots Z_n$$

Then,

$$I(g) \approx (b_1 - a_1)(b_2 - a_2)(b_3 - a_3)\frac{1}{n}\sum_{i=1}^{n} g(X_i, Y_i, Z_i). \tag{10.3}$$

Example 10.8. *Consider the function*

$$g(x, y, z) = \frac{32x^3}{3(y + z^4 + 1)}$$

with x, y and z all between 0 and 1.

To obtain a sense of the distribution of the approximations to the integral $I(g)$, we perform 1000 simulations using 100 uniform random variable for each of the three coordinates to perform the Monte Carlo integration. The command `Ig<-rep(0,1000)` *creates a vector of 1000 zeros. This is added so that* R *creates a place ahead of the simulations to store the results.*

```
> Ig<-rep(0,1000)
> for(i in 1:1000){x<-runif(100);y<-runif(100);z<-runif(100);
  g<-32*x^3/(3*(y+z^4+1)); Ig[i]<-mean(g)}
> hist(Ig)
> summary(Ig)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.045   1.507   1.644   1.650   1.788   2.284
> var(Ig)
[1] 0.03524665
> sd(Ig)
[1] 0.1877409
```

Thus, our Monte Carlo estimate the standard deviation of the estimated integral is 0.188.

Exercise 10.9. *Estimate the variance and standard deviation of the Monte Carlo estimator for the integral in the example above based on $n = 500$ and 1000 random numbers.*

Exercise 10.10. *How many observations are needed in estimating the integral in the example above so that the standard deviation of the average is 0.05?*

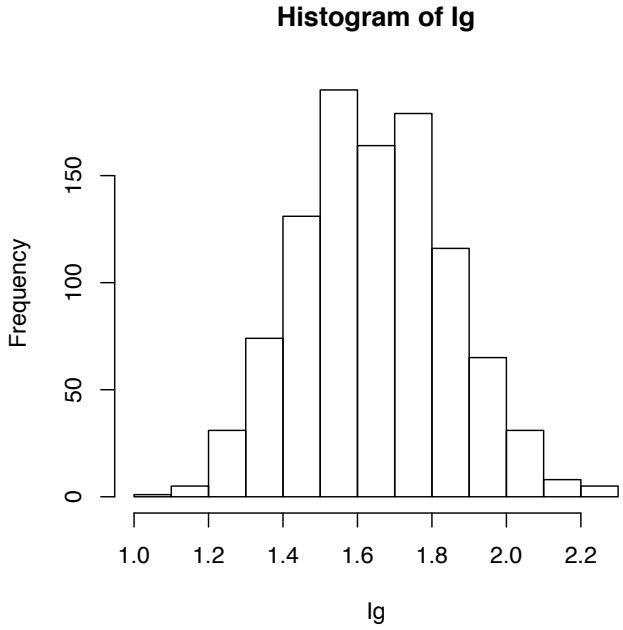

Figure 10.4: Histogram of 1000 Monte Carlo estimates for the integral $\int_0^1 \int_0^1 \int_0^1 32x^3/(y+z^4+1)\,dx\,dy\,dz$. The sample standard deviation is 0.188.

To modify this technique for a region $[a_1, b_1] \times [a_2, b_2] \times [a_3, b_3]$ use indepenent uniform random variables X_i, Y_i, and Z_i on the respective intervals, then

$$\frac{1}{n}\sum_{i=1}^{n} g(X_i, Y_i, Z_i) \approx Eg(X_1, Y_1, Z_1) = \frac{1}{b_1 - a_1}\frac{1}{b_2 - a_2}\frac{1}{b_3 - a_3}\int_{a_1}^{b_1}\int_{a_2}^{b_2}\int_{a_3}^{b_3} g(x, y, z)\,dz\,dy\,dx.$$

Thus, the estimate for the integral is

$$\frac{(b_1 - a_1)(b_2 - a_2)(b_3 - a_3)}{n}\sum_{i=1}^{n} g(X_i, Y_i, Z_i).$$

Exercise 10.11. *Use Monte Carlo integration to estimate*

$$\int_0^3 \int_{-2}^{2} \frac{\cos(\pi(x+y))}{\sqrt[4]{1+xy^2}} dydx.$$

10.3 Importance Sampling

In many of the large simulations, the dimension of the integral can be in the hundreds and the function g can be very close to zero for large regions in the domain of g. Simple Monte Carlo simulation will then frequently choose values for g that are close to zero. These values contribute very little to the average. Due to this inefficiency, a more sophisticated strategy is employed. **Importance sampling methods** begin with the observation that a better strategy may be to concentrate the random points where the integrand g is large in absolute value.

For example, for the integral

$$\int_0^1 \frac{e^{-x/2}}{\sqrt{x(1-x)}} dx, \tag{10.4}$$

the integrand is much bigger for values near $x = 0$ or $x = 1$. (See Figure 10.4) Thus, we can hope to have a more accurate estimate by concentrating our sample points in these places.

With this in mind, we perform the Monte Carlo integration beginning with $Y_1, Y_2, \ldots$ independent random variables with common densityf_Y. The goal is to find a density f_Y that is large when $|g|$ is large and small when $|g|$ is small. The density f_Y is called the **importance sampling function** or the **proposal density**. With this choice of density, we define the **importance sampling weights** so that

$$g(y) = w(y)f_Y(y). \tag{10.5}$$

To justify this choice, note that, the sample mean

$$\overline{w(Y)}_n = \frac{1}{n}\sum_{i=1}^{n} w(Y_i) \approx \int_{-\infty}^{\infty} w(y)f_Y(y)\, dy = \int_{-\infty}^{\infty} g(y)dy = I(g).$$

Thus, the average of the importance sampling weights, by the strong law of large numbers, still approximates the integral of g. This is an improvement over simple Monte Carlo integration if the variance decreases, i.e.,

$$\text{Var}(w(Y_1)) = \int_{-\infty}^{\infty} (w(y) - I(g))^2 f_Y(y)\, dy = \sigma_f^2 << \sigma^2.$$

As the formula shows, this can be best achieved by having the weight $w(y)$ be close to the integral $I(g)$. Referring to equation (10.5), we can now see that we should endeavor to have f_Y proportional to g.

Importance leads to the following procedure for estimating integrals.

- Write the integrand $g(x) = w(x)f_Y(x)$. Here f_Y is the density function for a random variable Y that is chosen to capture the changes in g.
- Simulate variables $Y_1, Y_2 \ldots, Y_n$ with density f_Y. This will sometimes require integrating the density function to obtain the distribution function $F_Y(x)$, and then finding its inverse function $F_Y^{-1}(u)$. This sets up the use of the probability transform to obtain $Y_i = F_Y^{-1}(U_i)$ where $U_1, U_2 \ldots, U_n$, independent random variables uniformly distributed on the interval $[0, 1]$.
- Compute the average of $w(Y_1), w(Y_2) \ldots, w(Y_n)$ to estimate the integral of g.

Note that the use of the probability transform removes the need to multiply $b - a$, the length of the interval.

Example 10.12. *For the integral (10.4) we can use Monte Carlo simulation based on uniform random variables.*

```
> Ig<-rep(0,1000)
> for(i in 1:1000){x<-runif(100);g<-exp(-x/2)*1/sqrt(x*(1-x));Ig[i]<-mean(g)}
> summary(Ig)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.970   2.277   2.425   2.484   2.583   8.586
> sd(Ig)
[1] 0.3938047
```

Based on a 1000 simulations, we find a sample mean value of 2.484 and a sample standard deviation of 0.394. Because the integrand is very large near $x = 0$ and $x = 1$, we choose look for a density f_Y to concentrate the random samples near the ends of the intervals.

Our choice for the proposal density is a $Beta(1/2, 1/2)$, then

$$f_Y(y) = \frac{1}{\pi} y^{1/2-1}(1-y)^{1/2-1}$$

on the interval $[0, 1]$. Thus the weight

$$w(y) = \pi e^{-y/2}$$

is the ratio $g(x)/f_Y(x)$

Again, we perform the simulation multiple times.

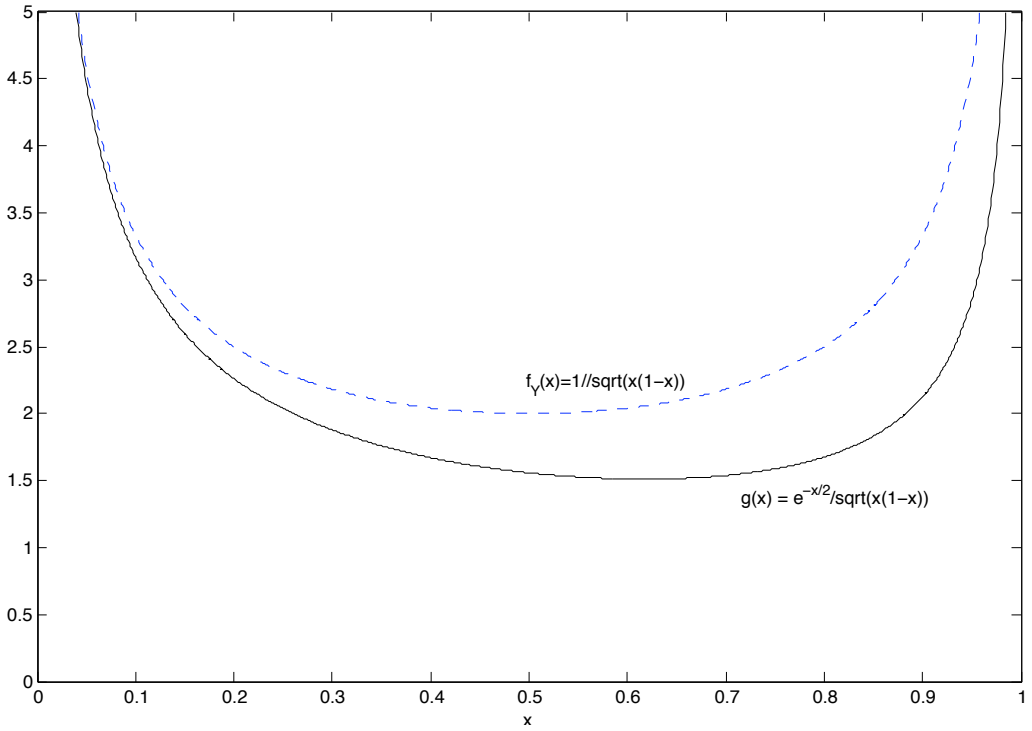

Figure 10.5: (left) Showing converges of the running average S_n/n to its limit 2 for $p = 1/2$. (right) Showing lack of convergence for the case $p = 2$.

```
> IS<-rep(0,1000)
> for(i in 1:1000){y<-rbeta(100,1/2,1/2);w<-pi*exp(-y/2);IS[i]<-mean(w)}
> summary(IS)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.321   2.455   2.483   2.484   2.515   2.609
> var(IS)
[1] 0.0002105915
> sd(IS)
[1] 0.04377021
```

Based on 1000 simulations, we find a sample mean value again of 2.484 and a sample standard deviation of 0.044, about 1/9th the size of the Monte Carlo weright. Part of the gain is illusory. Beta random variables take longer to simulate. If they require a factor more than 81 times longer to simulate, then the extra work needed to create a good importance sample is not helpful in producing a more accurate estimate for the integral. Numerical integration gives

```
> g<-function(x){exp(-x/2)*1/sqrt(x*(1-x))}
> integrate(g,0,1)
2.485054 with absolute error < 2e-06
```

Exercise 10.13. *Evaluate the integral*

$$\int_0^1 \frac{e^{-x}}{\sqrt[3]{x}}\,dx$$

1000 times using $n = 200$ *sample points using directly Monte Carlo integration and using importance sampling with random variables having density*

$$f_X(x) = \frac{2}{3\sqrt[3]{x}}$$

on the interval $[0, 1]$. *For the second part, you will need to use the probability transform. Compare the means and standard deviations of the 1000 estimates for the integral. The integral is approximately 1.04969.*

10.4 Answers to Selected Exercises

10.3. For $p \neq 1$, the expected value

$$EU^{-p} = \int_0^1 u^{-p}\, dp = \frac{1}{1-p} u^{1-p}\Big|_0^1 = \frac{1}{1-p} < \infty$$

provided that $1 - p > 0$ or $p < 1$. For $p > 1$, we evaluate the integral in the interval $[b, 1]$ and take a limit as $b \to 0$,

$$\int_b^1 u^{-p}\, dp = \frac{1}{1-p} u^{1-p}\Big|_b^1 = \frac{1}{1-p}(1 - b^{1-p}) \to \infty.$$

For $p = 1$,

$$\int_b^1 u^{-1}\, dp = \ln u\Big|_b^1 = -\ln b \to \infty.$$

We use the case $p = 1/2$ for which the integral converges. and $p = 2$ in which the integral does not. Indeed,

$$\int_)^1 u^{1/2}\, du = 2u^{3/2}\Big|_0^1 = 2$$

```
> par(mfrow=c(1,2))
> u<-runif(1000)
> x<-1/u^(1/2)
> s<-cumsum(x)
> plot(s/n,n,type="l")
> x<-1/u^2
> s<-cumsum(x)
> plot(n,s/n,type="l")
```

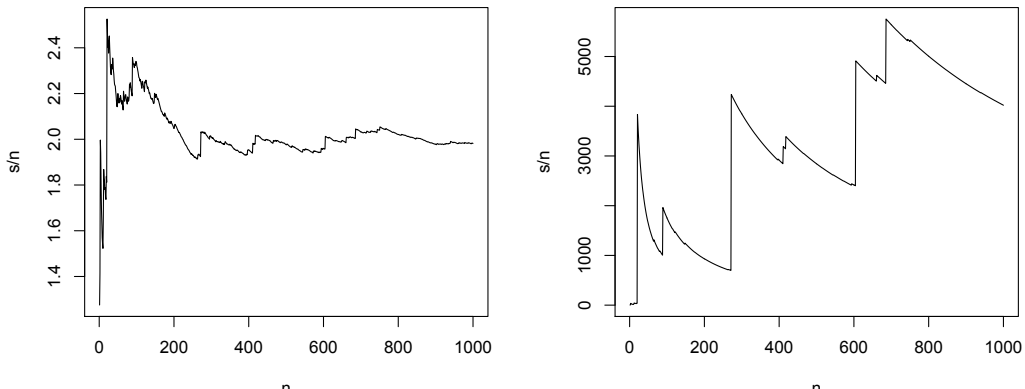

Figure 10.6: Importance sampling using the density function f_Y to estimate $\int_0^1 g(x)\, dx$. The weight $w(x) = \pi e^{-x/2}$.

10.5. Here are the R commands:

```
> par(mfrow=c(2,2))
> x<-rcauchy(1000)
> s<-cumsum(x)
```

```
> plot (n,s/n,type="l")
> x<-rcauchy(1000)
> s<-cumsum(x)
> plot (n,s/n,type="l")
> x<-rcauchy(1000)
> s<-cumsum(x)
> plot (n,s/n,type="l")
> x<-rcauchy(1000)
> s<-cumsum(x)
> plot (n,s/n,type="l")
```

This produces in Figure 10.5. Notice the differences for the values on the x-axis

10.9. The standard deviation for the average of n observations is $\sigma/\sqrt{n}$ where σ is the standard deviation for a single observation. From the output

```
> sd(Ig)
[1] 0.1877409
```

We have that $0.1877409 \approx \sigma/\sqrt{100} = \sigma/10$. Thus, $\sigma \approx 1.877409$. Consequently, for 500 observations, $\sigma/\sqrt{500} \approx 0.08396028$. For 1000 observations $\sigma/\sqrt{1000} \approx 0.05936889$

10.10. For $\sigma/\sqrt{n} = 0.05$, we have that $n = (\sigma/0.05)^2 \approx 1409.866$. So we need approximately 1410 observations.

10.11. To view the surface for $\frac{\cos(\pi(x+y))}{\sqrt[4]{1+xy^2}}$, $0 \leq x \leq 3$, $-2 \leq y \leq 2$, we type

```
> x <- seq(0,3, len=30)
> y <- seq(-2,2, len=30)
> f <- outer(x, y, function(x, y)
(cos(pi*(x+y)))/(1+x*y^2)^(1/4))
> persp(x,y,f,col="orange",phi=45,theta=30)
```

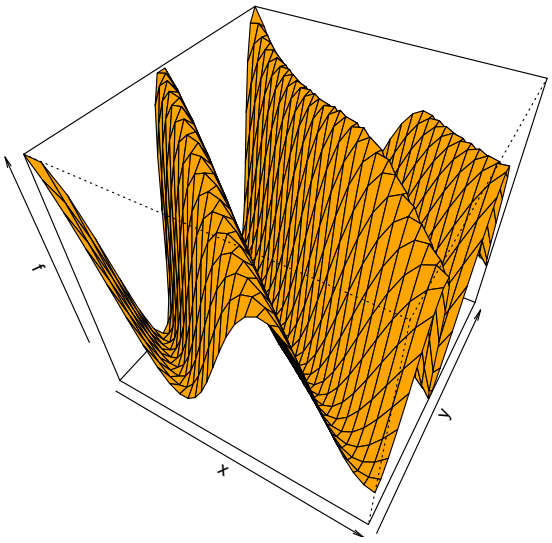

Figure 10.8: Surface plot of function used in Exercise 10.10.

Using 1000 random numbers uniformly distributed for both x and y, we have

```
> x<-runif(1000,0,3)
> y<-runif(1000,-2,2)
> g<-(cos(pi*(x+y)))/(1+x*y^2)^(1/4)
> 3*4*mean(g)
[1] 0.2452035
```

To finish, we need to multiply the average of g as estimated by `mean(g)` by the area associated to the integral $(3-0) \times (2-(-2)) = 12$.

10.13. For the direct Monte Carlo simulation, we have

```
> Ig<-rep(0,1000)
> for (i in 1:1000){x<-runif(200);g<-exp(-x)/x^(1/3);Ig[i]<-mean(g)}
> mean(Ig)
[1] 1.048734
> sd(Ig)
[1] 0.07062628
```

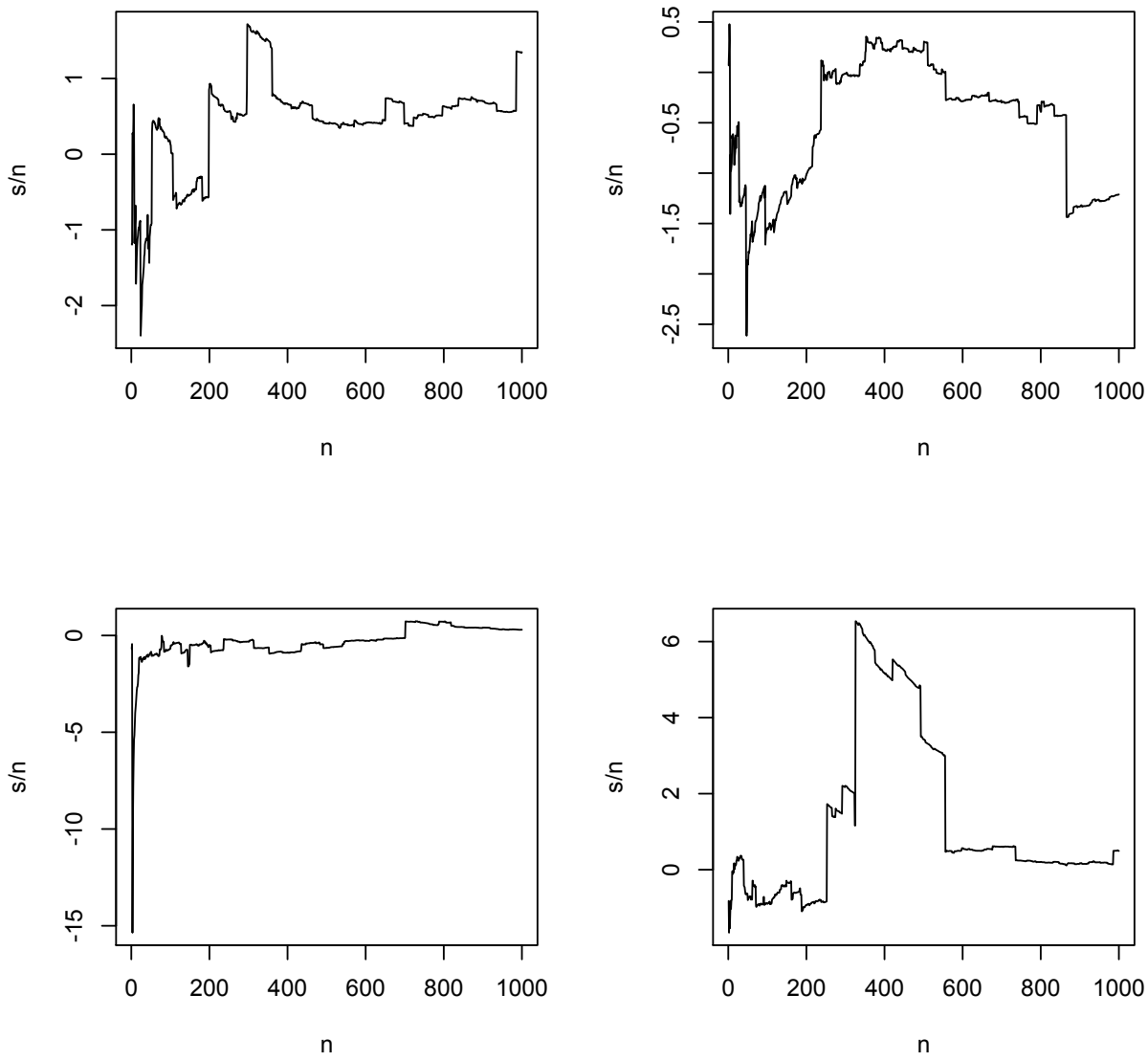

Figure 10.7: Plots of running averages of Cauchy random variables.

For the importance sampler, the integral is

$$\frac{3}{2}\int_0^1 e^{-x} f_X(x)\,dx.$$

To simulate independent random variables with density f_X, we first need the cumulative distribution function for X,

$$F_X(x) = \int_0^x \frac{2}{3\sqrt[3]{t}}\,dt = t^{2/3}\Big|_0^x = x^{2/3}.$$

Then, to find the probability transform, note that

$$u = F_X(x) = x^{2/3} \quad \text{and} \quad x = F_X^{-1}(u) = u^{3/2}.$$

Thus, to simulate X, we simulate a uniform random variable U on the interval $[0, 1]$ and evaluate $U^{3/2}$. This leads to the following R commands for the importance sample simulation:

```
> ISg<-rep(0,1000)
> for (i in 1:1000){u<-runif(200);x<-u^(3/2); w<-3*exp(-x)/2;ISg[i]<-mean(w)}
> mean(ISg)
[1] 1.048415
> sd(ISg)
[1] 0.02010032
```

Thus, the standard deviation using importance sampling is about 2/7-ths the standard deviation using simple Monte Carlo simulation. Consequently, we will can decrease the number of samples using importance sampling by a factor of $(2/7)^2 \approx 0.08$.