1.4 Monte Carlo integration

Monte Carlo integration is a widely used technique in many branches of mathematics and engineering and is conceptually very simple. Suppose the random variable X has arbitrary probability distribution $p(x)$ and we have an algorithm for generating a large number of independent realisations $x^{(1)}, x^{(2)}, ..., x^{(T)}$ from this distribution. Then

$$E(X) = \int xp(x)\,dx \approx \frac{1}{T}\sum_{t=1}^{T} x^{(t)}.$$

In other words, the theoretical expectation of X may be approximated by the sample mean of a set of independent realisations drawn from $p(x)$. By the Strong Law of Large Numbers, the approximation becomes arbitrarily exact as $T \to \infty$. Monte Carlo integration extends straightforwardly to the evaluation of more complex integrals. For example, the expectation of any function of X, $g(X)$, can be calculated as

$$E(g(X)) = \int g(x)p(x)\,dx \approx \frac{1}{T}\sum_{t=1}^{T} g(x^{(t)}),$$

that is, the sample mean of the functions of the simulated values. In particular, since the variance of X is simply a function of the expectations of X and X^2, this too may be approximated in a natural way using Monte Carlo integration. Not surprisingly, this estimate turns out to be the sample variance of the realisations $x^{(1)}, x^{(2)}, ..., x^{(T)}$ from $p(x)$.

Another important function of X is the indicator function, $I(l < X < u)$, which takes value 1 if X lies in the interval (l, u) and 0 otherwise. The expectation of $I(l < X < u)$ with respect to $p(x)$ gives the probability that X lies within the specified interval, $\Pr(l < X < u)$, and may be approximated using Monte Carlo integration by taking the sample average of the value of the indicator function for each realisation $x^{(t)}$. It is straightforward to see that this gives

$$\Pr(l < X < u) \approx \frac{\text{number of realisations } x^{(t)} \in (l, u)}{T}. \qquad (1.1)$$

In general, any desired summary of $p(x)$ may be approximated by calculating the corresponding summary of the sampled values generated from $p(x)$, with the approximation becoming increasingly exact as the sample size increases. Hence the theoretical quantiles of $p(x)$ may be estimated using the equivalent empirical quantile in the sample, and the shape of the density $p(x)$ may be approximated by constructing a histogram (or alternatively a "kernel density estimate" which effectively "smooths" the histogram) of the sampled values.

Suppose we obtain an empirical mean $\widehat{E} = \widehat{E}(g(X))$ and variance $\widehat{V} = \widehat{Var}(g(X))$ based on T simulated values, and we consider $\widehat{E}$ as the estimate of interest. Then, since $\widehat{E}$ is a sample mean based on T independent samples, it has true sample variance $Var(g(X))/T$, which may be estimated by $\widehat{V}/T$. Hence $\widehat{E}$ has an estimated standard error $\sqrt{\widehat{V}/T}$, which is known as the *Monte Carlo error*: see §4.5 for further discussion of this concept. We note that this may be reduced to any required degree of precision by increasing the number of simulated values.

Example 1.4.1. *Coins: a Monte Carlo approach to estimating tail areas*
Suppose we want to know the probability of getting 2 or fewer heads when we toss a fair coin 8 times. In formal terms, if $Y \sim \text{Binomial}(\pi, n)$, $\pi = 0.5, n = 8$, then what is $\Pr(Y \leq 2)$? We can identify four methods:

1. An *exact analytic* approach uses knowledge of the first three terms of the binomial distribution to give

$$\begin{aligned}\Pr(Y \leq 2) &= \sum_{y=0}^{2} p\left(y|\pi = 0.5, n = 8\right) \\ &= \binom{8}{0}\left(\frac{1}{2}\right)^{8}\left(\frac{1}{2}\right)^{0} + \binom{8}{1}\left(\frac{1}{2}\right)^{7}\left(\frac{1}{2}\right)^{1} + \binom{8}{2}\left(\frac{1}{2}\right)^{6}\left(\frac{1}{2}\right)^{2} \\ &= 0.1445.\end{aligned}$$

2. An *approximate analytic* approach might use our knowledge that $E[Y] = n\pi = 4$ and $Var[Y] = n\pi(1-\pi) = 2$ to create an approximate distribution $p(y) \approx \text{Normal}(4, 2)$, giving rise to an estimate of $\Pr(Y \leq 2) = \Phi((2 - 4)/\sqrt{2}) = 0.079$, or with a "continuity correction" $\Phi((2.5 - 4)/\sqrt{2}) = 0.144$; the latter is a remarkably good approximation.

3. A *physical* approach would be to repeatedly throw a set of 8 coins and count the proportion of trials where there were 2 or fewer heads. We did this 10 times, observed 0/10 cases of 2 or fewer heads, and then got bored!

4. A *simulation* approach uses a computer to toss the coins! Many programs have random number generators that produce an unstructured stream of numbers between 0 and 1. By checking whether each of these numbers lies above or below 0.5, we can simulate the toss of an individual fair coin, and by repeating in sets of 8 we can simulate the simultaneous toss of 8 coins. Figure 1.3 shows the empirical distributions after 100 and 10,000 trials and compares with the true binomial distribution. It is clear that extending the simulation improves the estimate of the required property of the underlying probability distribution.

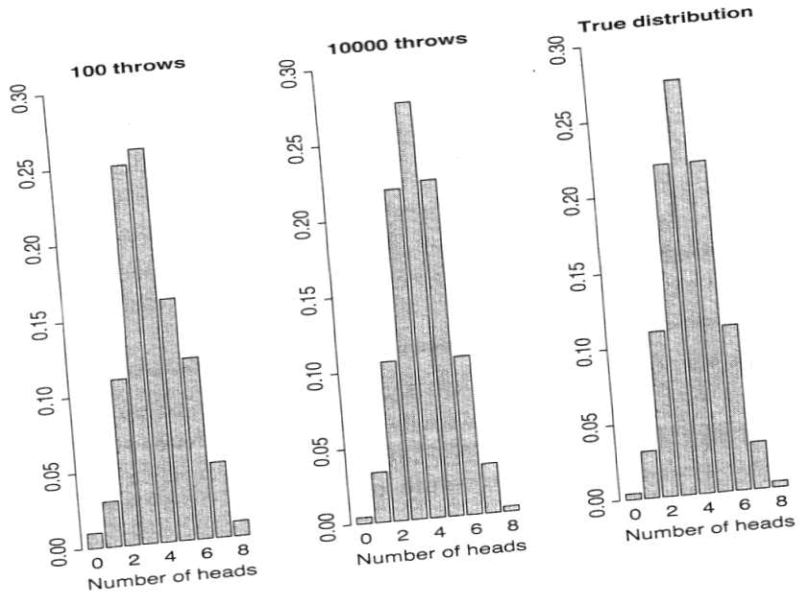

FIGURE 1.3
Distribution of the number of "heads" in trials of 8 tosses, from which we calculate the proportion with 2 or fewer heads: (a) after 100 trials (0.160); (b) after 10,000 trials (0.1450); (c) the true binomial distribution (0.1445).

Suppose we consider an indicator function $P2$ which takes on the value of 1 when there are 2 or fewer heads, 0 otherwise, so that $P2$ is a Bernoulli random quantity with expectation π, which we can calculate to be 0.1445, and true variance $\pi(1-\pi)$ =0.124. The true Monte Carlo error for an estimate of π based on T simulated values is therefore $\sqrt{\pi(1-\pi)/T}$, corresponding to the classical standard error of an estimate of π. Our estimates of π after 100 and 10,000 samples are 0.16 and 0.145, respectively, and so we can estimate Monte Carlo errors of 0.037 for $T = 100$ and 0.0035 for $T = 10,000$. If we took a classical statistical perspective we could therefore calculate approximate confidence intervals for π of $0.16 \pm 2 \times 0.037 = (0.09, 0.23)$ after 100 iterations, and $0.145 \pm 2 \times 0.0035 = (0.138, 0.152)$: both comfortably include the true value of 0.1445.

The above results are enormously useful, but to see the real beauty of Monte Carlo integration, suppose now that $\boldsymbol{X}$ is a random *vector* comprising k components, $X_1, ..., X_k$. Further suppose that $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, ..., \boldsymbol{x}^{(T)}$ are k-dimensional realisations, with elements denoted $x_j^{(i)}$ $(i = 1, ..., T,\ j = 1, ..., k)$, from the joint distribution $p(\boldsymbol{x})$. Then for any $j \in \{1, ..., k\}$, $x_j^{(1)}, x_j^{(2)}, ..., x_j^{(T)}$ represents a sample from $p(x_j)$. In other words, we can make inferences with respect to any marginal distribution by simply using those realisations that pertain to the random variable(s) of interest, and ignoring all others. This result holds for all possible marginal distributions, including those of arbitrary subsets of $\boldsymbol{X}$.

Such marginalisation, for example, integrating out of "nuisance" parameters, is a key component of modern Bayesian inference.

One could argue that the whole development of Bayesian analysis was delayed for decades due to lack of suitable computational tools, which explains why recent availability of high-performance personal computers has led to a revolution in simulation-based Bayesian methods.