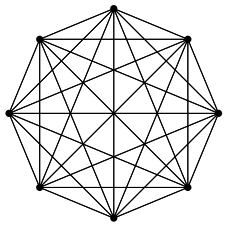
Chapter 19

Probability



R includes a group of functions that perform probability calculations that deserve a chapter of their own. This suite of well-organized functions are able to calculate certain quantities associated with the probability distributions of random variables. The topics considered in this chapter are (a) random numbers, (b) the binomial distribution, (c) the Poisson distribution, (d) the uniform distribution, (e) the normal distribution, (f) other distributions, and (g) random sampling. Two of these distributions are discrete distributions, namely the binomial and Poisson distributions, and two of these distributions are continuous distributions, namely the uniform and normal distributions. A rudimentary familiarity with probability is helpful for reading this chapter.

19.1 Random numbers

A *random number* is generally synonymous with a number that is uniformly distributed between 0 and 1. When the R function `runif` (the first letter `r` is short for random and the final letters `unif` are short for uniform) is called with a single integer argument `n`, it generates `n` random numbers. To generate a vector of four random numbers, for example, type

```
> runif(4)              # four U(0, 1) random numbers
[1] 0.2655087 0.3721239 0.5728534 0.9082078
```

All four of the numbers generated lie between 0 and 1. Although these four values appear to be independent and random, they are, in fact, generated by a deterministic algorithm within R (more details on this algorithm are given in Chapter 25). For this reason, they are called *pseudo-random numbers*. If the `runif` function is called again, four different random numbers are generated.

```
> runif(4)              # four more U(0, 1) random numbers
[1] 0.2016819 0.8983897 0.9446753 0.6607978
```

Situations frequently arise in which it is helpful to get the same set of random numbers in a subsequent call to `runif`. The `set.seed` function sets the random number seed for the generation of random numbers; it has an integer argument `seed` with a maximum value `.Machine$integer.max`, which is `2147483647`. A call to `set.seed` with an argument of 6 (the integer argument 6 was chosen arbitrarily), followed by a call to `runif(4)` yields

```
> set.seed(6)           # set the random number seed to 6
> runif(4)              # four more U(0, 1) random numbers
[1] 0.6062683 0.9376420 0.2643521 0.3800939
```

Now if `set.seed` is again called with an argument of 6, the random number stream has been reset to the same position as before, which means that the same random numbers will be generated.

```
> set.seed(6)           # set the random number seed to 6
> runif(4)              # the same four U(0, 1) random numbers
[1] 0.6062683 0.9376420 0.2643521 0.3800939
```

A random number by itself can be useful, but is even more useful when it is transformed to a *random variate* associated with a random variable coming from a particular probability distribution. A random variate is a realization of a random variable. Although random variates can be generated by hand, they are usually generated on a computer for efficiency. The next four sections illustrate how various aspects of a probability distribution can be calculated for the binomial, Poisson, uniform, and normal distributions.

19.2 Binomial distribution

The *binomial distribution* models the number of "successes" in n independent Bernoulli trials (each of which has probability of success p and probability of failure $1-p$), where n is a fixed positive integer. The definition of success is determined by the modeler. It could be passing an exam, making a free throw, or even a negative event such as getting the flu. A Bernoulli trial can be thought of as a single flip of a biased coin. When n Bernoulli trials are conducted, each with an identical probability of success, p, the entire experiment is known as a *binomial random experiment*, which satisfies the following criteria.

- The random experiment consists of n identical Bernoulli trials, where n is fixed.
- There are two possible outcomes for each Bernoulli trial, typically known generically as "success" and "failure."
- The Bernoulli trials are mutually independent.
- Each Bernoulli trial has an identical probability of success p.

The probability mass function $f(x) = P(X = x)$ of a binomial random variable X with parameters n and p is

$$f(x) = \binom{n}{x} p^x (1-p)^{n-x} \qquad x = 0, 1, 2, \ldots, n.$$

The syntax for four R functions that calculate various quantities associated with a binomial random variable X is given in the table below. The symbol $\sim$ is read "is distributed as."

function	**returned value for** $X \sim \text{binomial}(n, p)$
`dbinom(x, n, p)`	calculates the probability mass function $f(x) = P(X = x)$
`pbinom(x, n, p)`	calculates the cumulative distribution function $F(x) = P(X \leq x)$
`qbinom(u, n, p)`	calculates the quantile (percentile) $F^{-1}(u)$, for $0 < u < 1$
`rbinom(m, n, p)`	generates m binomial(n, p) random variates

The first example calculates the probability mass function (pmf) of a binomial random variable with parameters $n = 5$ and $p = 1/2$ for $x = 3$. One practical interpretation of the quantity calculated is that it is the probability of flipping exactly three heads ($x = 3$) in five tosses ($n = 5$) of a fair ($p = 1/2$) coin.

```
> dbinom(3, 5, 1 / 2)   # pmf at x = 3 for X ~ binomial(5, 1 / 2)
[1] 0.3125
```

This quantity could also have been calculated by hand with

$$f(3) = \binom{5}{3}\left(\frac{1}{2}\right)^3\left(1-\frac{1}{2}\right)^{5-3} = \frac{5!}{2!3!}\cdot\frac{1}{8}\cdot\frac{1}{4} = 10\cdot\frac{1}{32} = \frac{5}{16} = 0.3125.$$

The second example calculates the cumulative distribution function (cdf) of a binomial random variable with parameters $n = 5$ and $p = 1/2$ for $x = 3$. One practical interpretation of the quantity calculated is that it is the probability of flipping three or fewer heads ($x = 3$) in five tosses ($n = 5$) of a fair ($p = 1/2$) coin.

```
> pbinom(3, 5, 1 / 2)   # cdf at x = 3 for X ~ binomial(5, 1 / 2)
[1] 0.8125
```

This quantity could also have been calculated by hand with

$$\begin{aligned}
F(3) &= P(X \le 3) \\
&= P(X=0) + P(X=1) + P(X=2) + P(X=3) \\
&= \binom{5}{0}\left(\frac{1}{2}\right)^0\left(\frac{1}{2}\right)^5 + \binom{5}{1}\left(\frac{1}{2}\right)^1\left(\frac{1}{2}\right)^4 + \binom{5}{2}\left(\frac{1}{2}\right)^2\left(\frac{1}{2}\right)^3 + \binom{5}{3}\left(\frac{1}{2}\right)^3\left(\frac{1}{2}\right)^2 \\
&= \frac{1}{32} + \frac{5}{32} + \frac{10}{32} + \frac{10}{32} \\
&= \frac{26}{32} \\
&= \frac{13}{16} \\
&= 0.8125.
\end{aligned}$$

The last example concerning the binomial distribution involves generating random variates. The `rbinom` function can be used to generate, say, 12 random binomial variates with $n = 5$ and $p = 1/2$ with the R command

```
> rbinom(12, 5, 1 / 2)  # 12 random variates for X ~ binomial(5, 1 / 2)
 [1] 3 5 4 3 3 1 3 4 1 2 3 2
```

Each of the 12 random variates generated can be interpreted as a count of the random number of heads in five flips of a fair coin. These values must necessarily be integers that lie between 0 and 5 inclusive.

19.3 Poisson distribution

The Poisson distribution was introduced by French mathematician Simeon Poisson (1781–1840). There are two common ways to apply the Poisson distribution. First, the Poisson distribution can be used as an approximation to the binomial distribution, which is effective for large n and small p. Second, the Poisson distribution can be used to model the number of events that occur at random