3

Probability and distributions

The concepts of randomness and probability are central to statistics. It is an empirical fact that most experiments and investigations are not perfectly reproducible. The degree of irreproducibility may vary: Some experiments in physics may yield data that are accurate to many decimal places, whereas data on biological systems are typically much less reliable. However, the view of data as something coming from a statistical distribution is vital to understanding statistical methods. In this section, we outline the basic ideas of probability and the functions that R has for random sampling and handling of theoretical distributions.

3.1 Random sampling

Much of the earliest work in probability theory was about games and gambling issues, based on symmetry considerations. The basic notion then is that of a random sample: dealing from a well-shuffled pack of cards or picking numbered balls from a well-stirred urn.

In R, you can simulate these situations with the `sample` function. If you want to pick five numbers at random from the set `1:40`, then you can write

```
> sample(1:40,5)
[1]  4 30 28 40 13
```

P. Dalgaard, *Introductory Statistics with R*,
DOI: 10.1007/978-0-387-79054-1_3,

The first argument (`x`) is a vector of values to be sampled and the second (`size`) is the sample size. Actually, `sample(40,5)` would suffice since a single number is interpreted to represent the length of a sequence of integers.

Notice that the default behaviour of `sample` is *sampling without replacement.* That is, the samples will not contain the same number twice, and `size` obviously cannot be bigger than the length of the vector to be sampled. If you want sampling with replacement, then you need to add the argument `replace=TRUE`.

Sampling with replacement is suitable for modelling coin tosses or throws of a die. So, for instance, to simulate 10 coin tosses we could write

```
> sample(c("H","T"), 10, replace=T)
 [1] "T" "T" "T" "T" "T" "H" "H" "T" "H" "T"
```

In fair coin-tossing, the probability of heads should equal the probability of tails, but the idea of a random event is not restricted to symmetric cases. It could be equally well applied to other cases, such as the successful outcome of a surgical procedure. Hopefully, there would be a better than 50% chance of this. You can simulate data with nonequal probabilities for the outcomes (say, a 90% chance of success) by using the `prob` argument to `sample`, as in

```
> sample(c("succ", "fail"), 10, replace=T, prob=c(0.9, 0.1))
 [1] "succ" "succ" "succ" "succ" "succ" "succ" "succ" "succ"
 [9] "succ" "succ"
```

This may not be the best way to generate such a sample, though. See the later discussion of the binomial distribution.

3.2 Probability calculations and combinatorics

Let us return to the case of sampling without replacement, specifically `sample(1:40, 5)`. The probability of obtaining a given number as the first one of the sample should be $1/40$, the next one $1/39$, and so forth. The probability of a given sample should then be $1/(40 \times 39 \times 38 \times 37 \times 36)$. In R, use the `prod` function, which calculates the product of a vector of numbers

```
> 1/prod(40:36)
[1] 1.266449e-08
```

However, notice that this is the probability of getting given numbers in a given order. If this were a Lotto-like game, then you would rather be interested in the probability of guessing a given *set* of five numbers correctly. Thus you need also to include the cases that give the same numbers in a different order. Since obviously the probability of each such case is going to be the same, all we need to do is to figure out how many such cases there are and multiply by that. There are five possibilities for the first number, and for each of these there are four possibilities for the second, and so forth; that is, the number is $5 \times 4 \times 3 \times 2 \times 1$. This number is also written as 5! (5 *factorial*). So the probability of a "winning Lotto coupon" would be

```
> prod(5:1)/prod(40:36)
[1] 1.519738e-06
```

There is another way of arriving at the same result. Notice that since the actual set of numbers is immaterial, all sets of five numbers must have the same probability. So all we need to do is to calculate the number of ways to choose 5 numbers out of 40. This is denoted

$$\binom{40}{5} = \frac{40!}{5!35!} = 658008$$

In R, the `choose` function can be used to calculate this number, and the probability is thus

```
> 1/choose(40,5)
[1] 1.519738e-06
```

3.3 Discrete distributions

When looking at independent replications of a binary experiment, you would not usually be interested in whether each case is a success or a failure but rather in the total number of successes (or failures). Obviously, this number is random since it depends on the individual random outcomes, and it is consequently called a *random variable*. In this case it is a discrete-valued random variable that can take values $0, 1, \ldots, n$, where n is the number of replications. Continuous random variables are encountered later.

A random variable X has a *probability distribution* that can be described using *point probabilities* $f(x) = P(X = x)$ or the *cumulative distribution function* $F(x) = P(X \leq x)$. In the case at hand, the distribution can be worked out as having the point probabilities

$$f(x) = \binom{n}{x} p^x (1-p)^{n-x}$$

This is known as the *binomial distribution*, and the $\binom{n}{x}$ are known as *binomial coefficients*. The parameter p is the probability of a successful outcome in an individual trial. A graph of the point probabilities of the binomial distribution appears in Figure 3.2 ahead.

We delay describing the R functions related to the binomial distribution until we have discussed continuous distributions so that we can present the conventions in a unified manner.

Many other distributions can be derived from simple probability models. For instance, the *geometric distribution* is similar to the binomial distribution but records the number of failures that occur before the first success.

3.4 Continuous distributions

Some data arise from measurements on an essentially continuous scale, for instance temperature, concentrations, etc. In practice, they will be recorded to a finite precision, but it is useful to disregard this in the modelling. Such measurements will usually have a component of random variation, which makes them less than perfectly reproducible. However, these random fluctuations will tend to follow patterns; typically they will cluster around a central value, with large deviations being more rare than smaller ones.

In order to model continuous data, we need to define random variables that can obtain the value of any real number. Because there are infinitely many numbers infinitely close, the probability of any particular value will be zero, so there is no such thing as a point probability as for discrete-valued random variables. Instead we have the concept of a *density*. This is the infinitesimal probability of hitting a small region around x divided by the size of the region. The cumulative distribution function can be defined as before, and we have the relation

$$F(x) = \int_{-\infty}^{x} f(x)\,dx$$

There are a number of standard distributions that come up in statistical theory and are available in R. It makes little sense to describe them in detail here except for a couple of examples.

The *uniform distribution* has a constant density over a specified interval (by default $[0, 1]$).

The *normal distribution* (also known as the *Gaussian distribution*) has density

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

depending on its mean μ and standard deviation σ. The normal distribution has a characteristic bell shape (Figure 3.1), and modifying μ and σ simply translates and widens the distribution. It is a standard building block in statistical models, where it is commonly used to describe error variation. It also comes up as an approximating distribution in several contexts; for instance, the binomial distribution for large sample sizes can be well approximated by a suitably scaled normal distribution.

3.5 The built-in distributions in R

The standard distributions that turn up in connection with model building and statistical tests have been built into R, and it can therefore completely replace traditional statistical tables. Here we look only at the normal distribution and the binomial distribution, but other distributions follow exactly the same pattern.

Four fundamental items can be calculated for a statistical distribution:

- Density or point probability
- Cumulated probability, distribution function
- Quantiles
- Pseudo-random numbers

For all distributions implemented in R, there is a function for each of the four items listed above. For example, for the normal distribution, these are named `dnorm`, `pnorm`, `qnorm`, and `rnorm` (*d*ensity, *p*robability, *q*uantile, and *r*andom, respectively).

3.5.1 Densities

The density for a continuous distribution is a measure of the relative probability of "getting a value close to x". The probability of getting a value in a particular interval is the area under the corresponding part of the curve.

For discrete distributions, the term "density" is used for the point probability — the probability of getting exactly the value x. Technically, this is correct: It is a density with respect to counting measure.

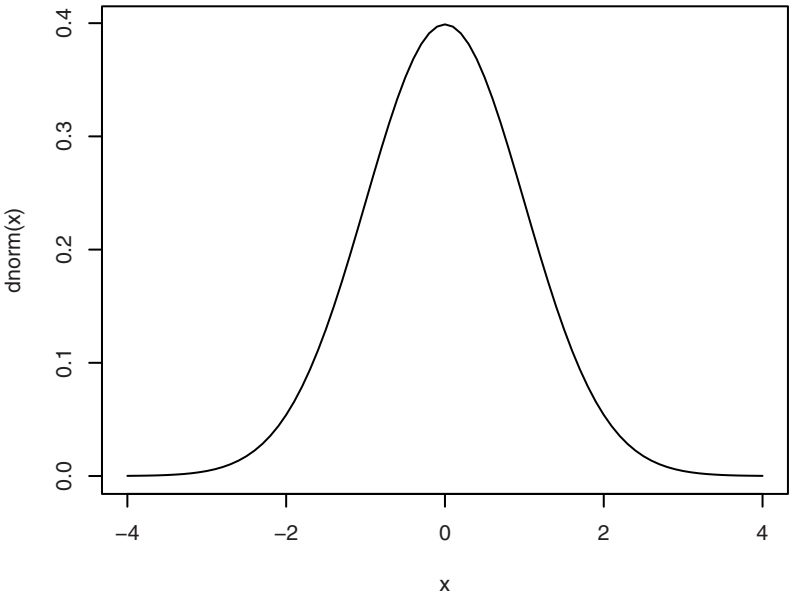

Figure 3.1. Density of normal distribution.

The density function is likely the one of the four function types that is least used in practice, but if for instance it is desired to draw the well-known bell curve of the normal distribution, then it can be done like this:

```
> x <- seq(-4,4,0.1)
> plot(x,dnorm(x),type="l")
```

(Notice that this is the letter 'l', not the digit '1').

The function `seq` (see p. 15) is used to generate equidistant values, here from -4 to 4 in steps of 0.1; that is, $(-4.0, -3.9, -3.8, \ldots, 3.9, 4.0)$. The use of `type="l"` as an argument to `plot` causes the function to draw lines between the points rather than plotting the points themselves.

An alternative way of creating the plot is to use `curve` as follows:

```
> curve(dnorm(x), from=-4, to=4)
```

This is often a more convenient way of making graphs, but it does require that the y-values can be expressed as a simple functional expression in x.

For discrete distributions, where variables can take on only distinct values, it is preferable to draw a pin diagram, here for the binomial distribution with $n = 50$ and $p = 0.33$ (Figure 3.2):

```
> x <- 0:50
> plot(x,dbinom(x,size=50,prob=.33),type="h")
```

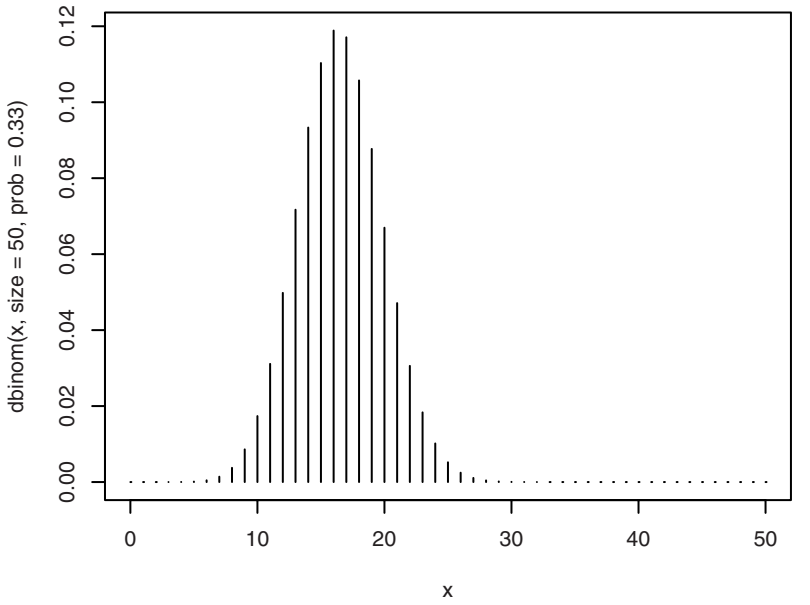

Figure 3.2. Point probabilities in binom(50, 0.33).

Notice that there are three arguments to the "d-function" this time. In addition to x, you have to specify the number of trials n and the probability parameter p. The distribution drawn corresponds to, for example, the number of 5s or 6s in 50 throws of a symmetrical die. Actually, `dnorm` also takes more than one argument, namely the mean and standard deviation, but they have default values of 0 and 1, respectively, since most often it is the standard normal distribution that is requested.

The form `0:50` is a short version of `seq(0,50,1)`: the whole numbers from 0 to 50 (see p. 15). It is `type="h"` (as in *h*istogram-like) that causes the pins to be drawn.

3.5.2 *Cumulative distribution functions*

The cumulative distribution function describes the probability of "hitting" x or less in a given distribution. The corresponding R functions begin with a 'p' (for probability) by convention.

Just as you can plot densities, you can of course also plot cumulative distribution functions, but that is usually not very informative. More often, actual numbers are desired. Say that it is known that some biochemical measure in healthy individuals is well described by a normal distribution with a mean of 132 and a standard deviation of 13. Then, if a patient has a value of 160, there is

```
> 1-pnorm(160,mean=132,sd=13)
[1] 0.01562612
```

or only about 1.5% of the general population, that has that value or higher. The function `pnorm` returns the probability of getting a value smaller than its first argument in a normal distribution with the given mean and standard deviation.

Another typical application occurs in connection with statistical tests. Consider a simple sign test: Twenty patients are given two treatments each (blindly and in randomized order) and then asked whether treatment A or B worked better. It turned out that 16 patients liked A better. The question is then whether this can be taken as sufficient evidence that A actually is the better treatment or whether the outcome might as well have happened by chance even if the treatments were equally good. If there was no difference between the two treatments, then we would expect the number of people favouring treatment A to be binomially distributed with $p = 0.5$ and $n = 20$. How (im)probable would it then be to obtain what we have observed? As in the normal distribution, we need a tail probability, and the immediate guess might be to look at

```
> pbinom(16,size=20,prob=.5)
[1] 0.9987116
```

and subtract it from 1 to get the upper tail — but this would be an error! What we need is the probability of *the observed or more extreme*, and `pbinom` is giving the probability of 16 or less. We need to use "15 or less" instead.

```
> 1-pbinom(15,size=20,prob=.5)
[1] 0.005908966
```

If you want a two-tailed test because you have no prior idea about which treatment is better, then you will have to add the probability of obtaining equally extreme results in the opposite direction. In the present case, that

means the probability that four or fewer people prefer A, giving a total probability of

```
> 1-pbinom(15,20,.5)+pbinom(4,20,.5)
[1] 0.01181793
```

(which is obviously exactly twice the one-tailed probability).

As can be seen from the last command, it is not strictly necessary to use the `size` and `prob` keywords as long as the arguments are given in the right order (positional matching; see Section 1.2.2).

It is quite confusing to keep track of whether or not the observation itself needs to be counted. Fortunately, the function `binom.test` keeps track of such formalities and performs the correct binomial test. This is further discussed in Chapter 8.

3.5.3 *Quantiles*

The quantile function is the inverse of the cumulative distribution function. The p-quantile is the value with the property that there is probability p of getting a value less than or equal to it. The median is by definition the 50% quantile.

Some details concerning the definition in the case of discontinuous distributions are glossed over here. You can fairly easily deduce the behaviour by experimenting with the R functions.

Tables of statistical distributions are almost always given in terms of quantiles. For a fixed set of probabilities, the table shows the boundary that a test statistic must cross in order to be considered significant at that level. This is purely for operational reasons; it is almost superfluous when you have the option of computing p exactly.

Theoretical quantiles are commonly used for the calculation of confidence intervals and for power calculations in connection with designing and dimensioning experiments (see Chapter 9). A simple example of a confidence interval can be given here (see also Chapter 5).

If we have n normally distributed observations with the same mean μ and standard deviation σ, then it is known that the average $\bar{x}$ is normally distributed around μ with standard deviation $\sigma/\sqrt{n}$. A 95% confidence interval for μ can be obtained as

$$\bar{x} + \sigma/\sqrt{n} \times N_{0.025} \leq \mu \leq \bar{x} + \sigma/\sqrt{n} \times N_{0.975}$$

where $N_{0.025}$ is the 2.5% quantile in the normal distribution. If $\sigma = 12$ and we have measured $n = 5$ persons and found an average of $\bar{x} = 83$, then

we can compute the relevant quantities as ("sem" means *standard error of the mean*)

```
> xbar <- 83
> sigma <- 12
> n <- 5
> sem <- sigma/sqrt(n)
> sem
[1] 5.366563
> xbar + sem * qnorm(0.025)
[1] 72.48173
> xbar + sem * qnorm(0.975)
[1] 93.51827
```

and thus find a 95% confidence interval for μ going from 72.48 to 93.52. (Notice that this is based on the assumption that σ is known. This is sometimes reasonable in process control applications. The more common case of estimating σ from the data leads to confidence intervals based on the t distribution and is discussed in Chapter 5.)

Since it is known that the normal distribution is symmetric, so that $N_{0.025} = -N_{0.975}$, it is common to write the formula for the confidence interval as $\bar{x} \pm \sigma/\sqrt{n} \times N_{0.975}$. The quantile itself is often written $\Phi^{-1}(0.975)$, where Φ is standard notation for the cumulative distribution function of the normal distribution (`pnorm`).

Another application of quantiles is in connection with Q–Q plots (see Section 4.2.3), which can be used to assess whether a set of data can reasonably be assumed to come from a given distribution.

3.5.4 Random numbers

To many people, it sounds like a contradiction in terms to generate random numbers on a computer since its results are supposed to be predictable and reproducible. What is in fact possible is to generate sequences of "pseudo-random" numbers, which for practical purposes behave *as if* they were drawn randomly.

Here random numbers are used to give the reader a feeling for the way in which randomness affects the quantities that can be calculated from a set of data. In professional statistics, they are used to create simulated data sets in order to study the accuracy of mathematical approximations and the effect of assumptions being violated.

The use of the functions that generate random numbers is straightforward. The first argument specifies the number of random numbers to compute, and the subsequent arguments are similar to those for other functions related to the same distributions. For instance,

```
> rnorm(10)
 [1] -0.2996466 -0.1718510 -0.1955634  1.2280843 -2.6074190
 [6] -0.2999453 -0.4655102 -1.5680666  1.2545876 -1.8028839
> rnorm(10)
 [1]  1.7082495  0.1432875 -1.0271750 -0.9246647  0.6402383
 [6]  0.7201677 -0.3071239  1.2090712  0.8699669  0.5882753
> rnorm(10,mean=7,sd=5)
 [1]  8.934983  8.611855  4.675578  3.670129  4.223117  5.484290
 [7] 12.141946  8.057541 -2.893164 13.590586
> rbinom(10,size=20,prob=.5)
 [1] 12 11 10  8 11  8 11  8  8 13
```

3.6 Exercises

3.1 Calculate the probability for each of the following events: (a) A standard normally distributed variable is larger than 3. (b) A normally distributed variable with mean 35 and standard deviation 6 is larger than 42. (c) Getting 10 out of 10 successes in a binomial distribution with probability 0.8. (d) $X < 0.9$ when X has the standard uniform distribution. (e) $X > 6.5$ in a χ^2 distribution with 2 degrees of freedom.

3.2 A rule of thumb is that 5% of the normal distribution lies outside an interval approximately $\pm 2s$ about the mean. To what extent is this true? Where are the limits corresponding to 1%, 0.5%, and 0.1%? What is the position of the quartiles measured in standard deviation units?

3.3 For a disease known to have a postoperative complication frequency of 20%, a surgeon suggests a new procedure. He tests it on 10 patients and there are no complications. What is the probability of operating on 10 patients successfully with the traditional method?

3.4 Simulated coin-tossing can be done using `rbinom` instead of `sample`. How exactly would you do that?