Continuous Probability Distributions

R has a wide range of built-in probability distributions, for each of which four functions are available: the probability density function (which has a d prefix); the cumulative probability (p); the quantiles of the distribution (q); and random numbers generated from the distribution (r). Each letter can be prefixed to the R function names in Table 7.1 (e.g. dbeta).

Table 7.1. The probability distributions supported by R. The meanings of the parameters are explained in the text.

R function	Distribution	Parameters
beta	beta	shape1, shape2
binom	binomial	sample size, probability
cauchy	Cauchy	location, scale
exp	exponential	rate (optional)
chisq	chi-squared	degrees of freedom
f	Fisher's *F*	df1, df2
gamma	gamma	shape
geom	geometric	probability
hyper	hypergeometric	*m, n, k*
lnorm	lognormal	mean, standard deviation
logis	logistic	location, scale
nbinom	negative binomial	size, probability
norm	normal	mean, standard deviation
pois	Poisson	mean
signrank	Wilcoxon signed rank statistic	sample size *n*
t	Student's *t*	degrees of freedom
unif	uniform	minimum, maximum (opt.)
weibull	Weibull	shape
wilcox	Wilcoxon rank sum	*m, n*

The **cumulative probability** function is a straightforward notion: it is an S-shaped curve showing, for any value of x, the probability of obtaining a sample value that is less than or equal to x. Here is what it looks like for the normal distribution:

```
curve(pnorm(x),-3,3)
arrows(-1,0,-1,pnorm(-1),col="red")
arrows(-1,pnorm(-1),-3,pnorm(-1),col="green")
```

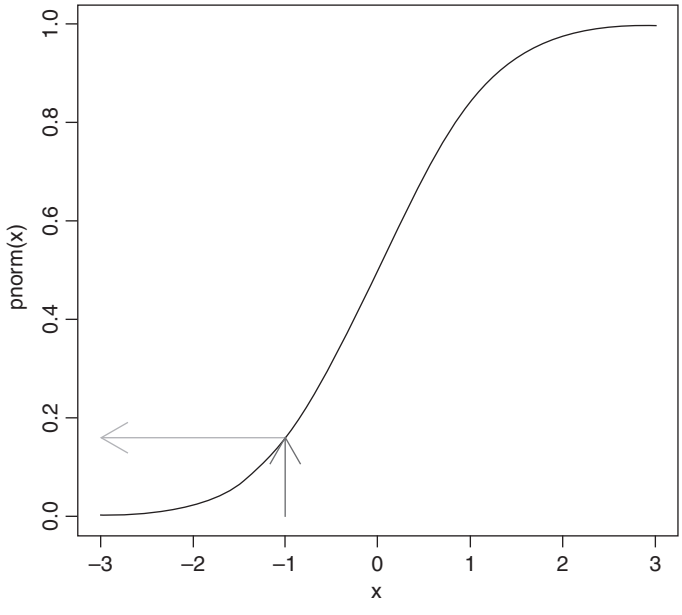
The value of $x(-1)$ leads up to the cumulative probability (red arrow) and the probability associated with obtaining a value of this size (-1) or smaller is on the y axis (green arrow). The value on the y axis is 0.158 655 3:

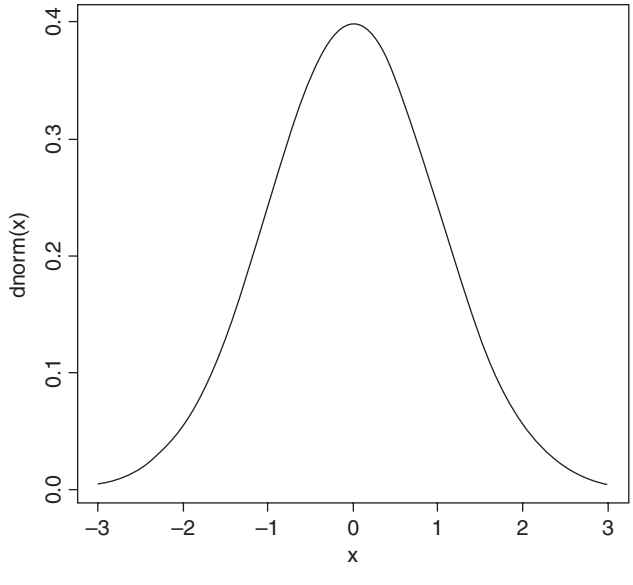
```
pnorm(-1)
```

```
[1]  0.1586553
```

The **probability density** is the slope of this curve (its 'derivative'). You can see at once that the slope is never negative. The slope starts out very shallow up to about $x = -2$, increases up to a peak (at $x = 0$ in this example) then gets shallower, and becomes very small indeed above about $x = 2$. Here is what the density function of the normal (dnorm) looks like:

```
curve(dnorm(x),-3,3)
```





For a discrete random variable, like the Poisson or the binomial, the probability density function is straightforward: it is simply a histogram with the *y* axis scaled as probabilities rather than counts, and the discrete values of *x* (0, 1, 2, 3, ...) on the horizontal axis. But for a continuous random variable, the definition of the probability density function is more subtle: it does not have probabilities on the *y* axis, but rather the derivative (the slope) of the cumulative probability function at a given value of *x*.

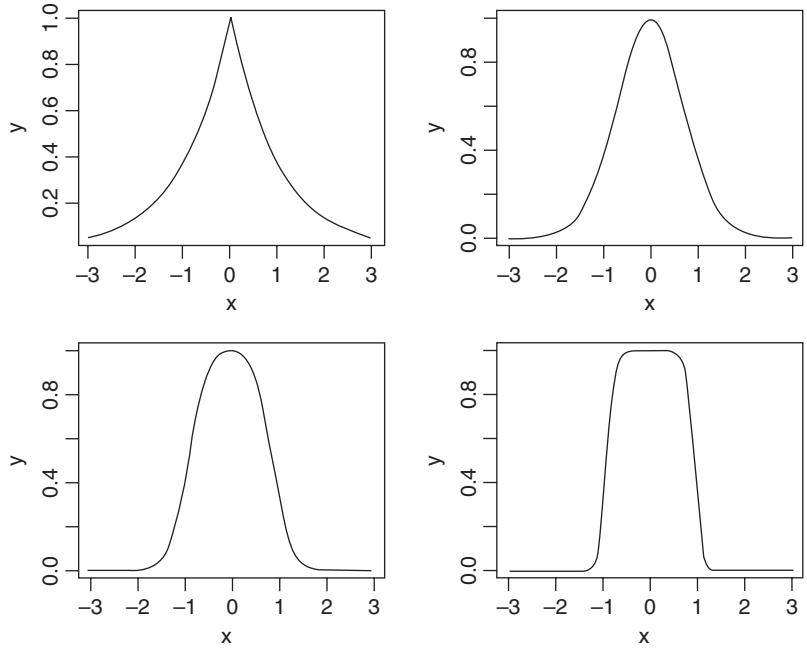
Normal distribution

This distribution is central to the theory of parametric statistics. Consider the following simple exponential function:

$$y = \exp(-|x|^m).$$

As the power (m) in the exponent increases, the function becomes more and more like a step function. The following panels show the relationship between y and x for $m = 1, 2, 3$ and 8, respectively:

```
par(mfrow=c(2,2))
x<-seq(-3,3,0.01)
y<-exp(-abs(x))
plot(x,y,type="l")
y<-exp(-abs(x)^2)
plot(x,y,type="l")
y<-exp(-abs(x)^3)
plot(x,y,type="l")
y<-exp(-abs(x)^8)
plot(x,y,type="l")
```



The second of these panels (top right), where $y = \exp(-x^2)$, is the basis of an extremely important and famous probability density function. Once it has been scaled, so that the integral (the area under the curve from $-\infty$ to $+\infty$) is unity, this is the normal distribution.

Unfortunately, the scaling constants are rather cumbersome. When the distribution has mean 0 and standard deviation 1 (the **standard normal** distribution) the equation becomes:

$$f(z) = \frac{1}{\sqrt{2\pi}} e^{-z^2/2}.$$

Suppose we have measured the heights of 100 people. The mean height was 170 cm and the standard deviation was 8 cm (top left panel, below). We can ask three sorts of questions about data like these: what is the probability that a randomly selected individual will be:

- shorter than a particular height?
- taller than a particular height?
- between one specified height and another?

The area under the whole curve is exactly 1; everybody has a height between minus infinity and plus infinity. True, but not particularly helpful. Suppose we want to know the probability that one of our people, selected at random from the group, will be less than 160 cm tall. We need to convert this height into a value of z; that is to say, we need to convert 160 cm into *a number of standard deviations from the mean*. What do we know about the standard normal distribution? It has a mean of 0 and a standard deviation of 1. So we can convert any value y, from a distribution with mean $\bar{y}$ and standard deviation s very simply by calculating:

$$z = \frac{y - \bar{y}}{s}.$$

So we convert 160 cm into a number of standard deviations. It is less than the mean height (170 cm) so its value will be negative:

$$z = \frac{160 - 170}{8} = -1.25.$$

Now we need to find the probability of a value of the standard normal taking a value of -1.25 or smaller. This is *the area under the left hand tail* (the integral) of the density function. The function we need for this is pnorm: we provide it with a value of z (or, more generally, with a quantile) and it provides us with the probability we want:

```
pnorm(-1.25)
```

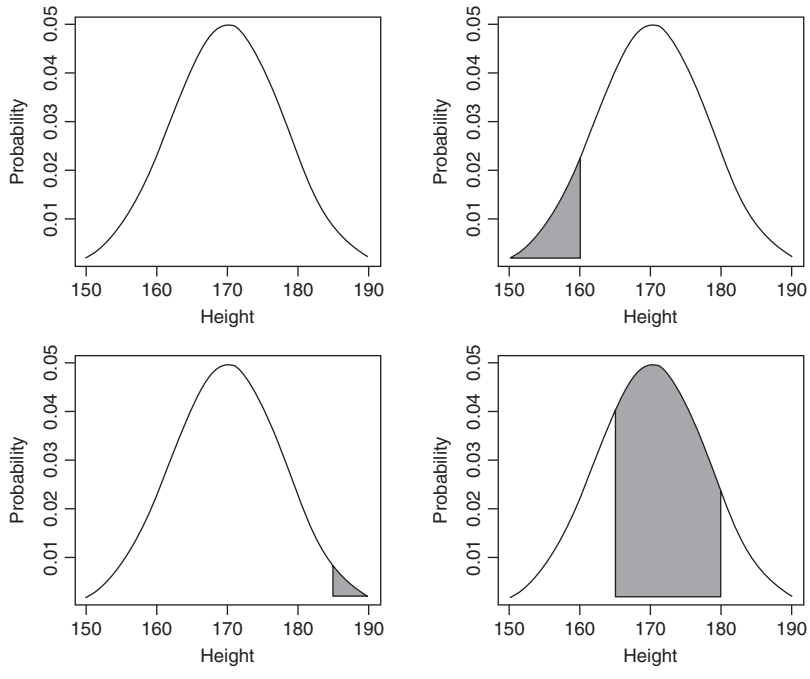
```
[1] 0.1056498
```

So the answer to our first question (the red area, top right) is just over 10%.

Next, what is the probability of selecting one of our people and finding that they are taller than 185 cm (bottom left)? The first two parts of the exercise are exactly the same as before. First we convert our value of 185 cm into a number of standard deviations:

$$z = \frac{185 - 170}{8} = 1.875.$$

Then we ask what probability is associated with this, using pnorm:




```
pnorm(1.875)
```

```
[1] 0.9696036
```

But this is the answer to a different question. This is the probability that someone will be *less* than or equal to 185 cm tall (that is what the function pnorm has been written to provide). All we need to do is to work out the complement of this:

```
1-pnorm(1.875)
```

```
[1] 0.03039636
```

So the answer to the second question is about 3%.

Finally, we might want to know the probability of selecting a person between 165 cm and 180 cm. We have a bit more work to do here, because we need to calculate two z values:

$$z_1 = \frac{165 - 170}{8} = -0.625 \quad \text{and} \quad z_2 = \frac{(180 - 170)}{8} = 1.25.$$

The important point to grasp is this: we want the probability of selecting a person between these two z values, so we *subtract the smaller probability from the larger probability*:

```
pnorm(1.25)-pnorm(-0.625)
```

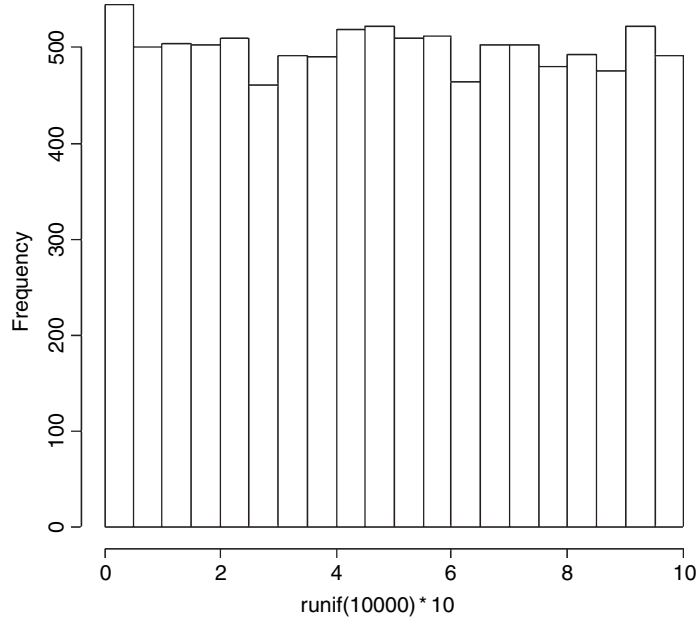
```
[1] 0.6283647
```

Thus we have a 63% chance of selecting a medium-sized person (taller than 165 cm and shorter than 180 cm) from this sample with a mean height of 170 cm and a standard deviation of 8 cm (bottom right, above).

The central limit theorem

If you take repeated samples from a population with finite variance and calculate their averages, then the averages will be normally distributed. This is called the **central limit theorem**. Let's demonstrate it for ourselves. We can take five uniformly distributed random numbers between 0 and 10 and work out the average. The average will be low when we get, say, 2,3,1,2,1 and big when we get 9,8,9,6,8. Typically, of course, the average will be close to 5. Let's do this 10 000 times and look at the distribution of the 10 000 means. The data are rectangularly (uniformly) distributed on the interval 0 to 10, so the distribution of the raw data should be flat-topped:

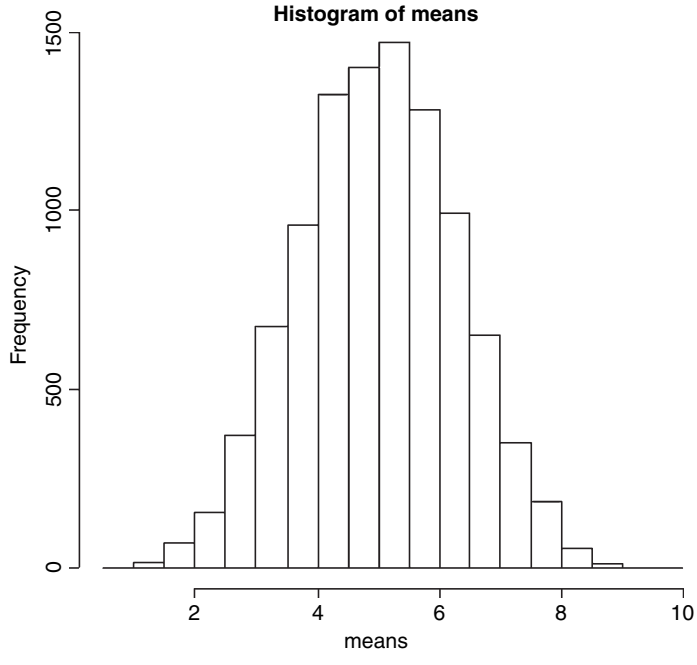
```
hist(runif(10000)*10,main="")
```



What about the distribution of sample means, based on taking just 5 uniformly distributed random numbers?

```
means<-numeric(10000)
for (i in 1:10000){
means[i]<-mean(runif(5)*10)
}
hist(means,ylim=c(0,1600))
```

Nice, but how close is this to a normal distribution? One test is to draw a normal distribution with the same parameters on top of the histogram. But what are these parameters? The normal is a two-parameter distribution that is characterized by its mean and its standard deviation. We can estimate these two parameters from our sample of 10 000 means (your values will be slightly different because of the randomization):




```
mean(means)
```

```
[1] 4.998581
```

```
sd(means)
```

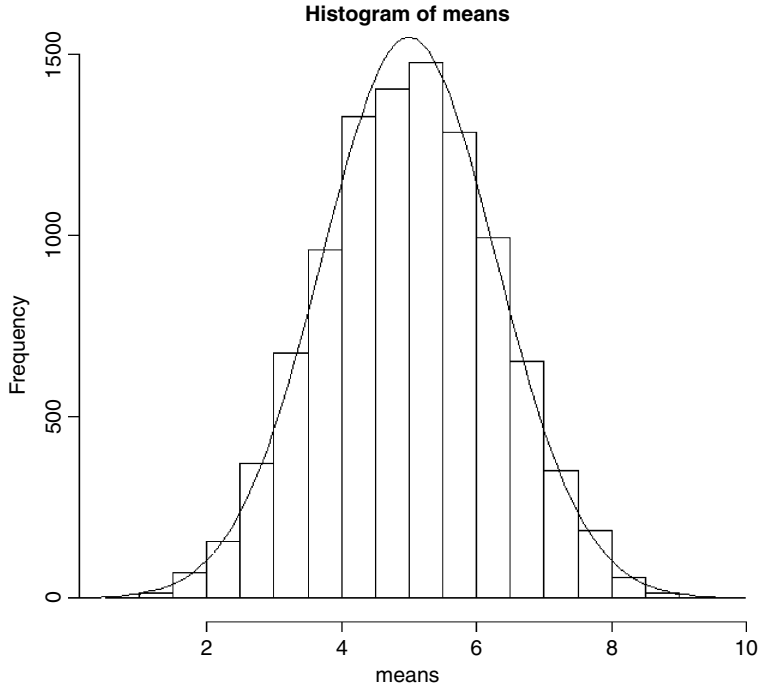
```
[1] 1.289960
```

Now we use these two parameters in the probability density function of the normal distribution (dnorm) to create a normal curve with our particular mean and standard deviation. To draw the smooth line of the normal curve, we need to generate a series of values for the *x* axis; inspection of the histograms suggest that sensible limits would be from 0 to 10 (the limits we chose for our uniformly distributed random numbers). A good rule of thumb is that for a smooth curve you need at least 100 values, so let's try this:

```
xv<-seq(0,10,0.1)
```

There is just one thing left to do. The probability density function has an integral of 1.0 (that's the area beneath the normal curve), but we had 10 000 samples. To scale the normal probability density function to our particular case, however, depends on the height of the highest bar (about 1500 in this case). The height, in turn, depends on the chosen bin widths; if we doubled with width of the bin there would be roughly twice as many numbers in the bin and the bar would be twice as high on the *y* axis. To get the height of the bars on our frequency scale, therefore, we multiply the total frequency, 10 000 by the bin width, 0.5 to get 5000. We multiply 5000 by the probability density to get the height of the curve. Finally, we use lines to overlay the smooth curve on our histogram:

```
yv<-dnorm(xv,mean=4.998581,sd=1.28996)*5000
lines(xv,yv)
```



The fit is excellent. The central limit theorem really works. Almost any distribution, even a 'badly behaved' one like the uniform distribution we worked with here, will produce a normal distribution of sample means taken from it.

A simple example of the operation of the central limit theorem involves the use of dice. Throw one die lots of times and each of the six numbers should come up equally often: this is an example of a **uniform** distribution:

```
par(mfrow=c(2,2))
hist(sample(1:6,replace=T,10000),breaks=0.5:6.5,main="",xlab="one die")
```

Now throw two dice and add the scores together: this is the ancient game of craps. There are 11 possible scores from a minimum of 2 to a maximum of 12. The most likely score is 7 because there are 6 ways that this could come about:

1,6 6,1 2,5 5,2 3,4 4,3

For many throws of craps we get a **triangular** distribution of scores, centred on 7:

```
a<-sample(1:6,replace=T,10000)
b<-sample(1:6,replace=T,10000)
hist(a+b,breaks=1.5:12.5,main="", xlab="two dice")
```

There is already a clear indication of central tendency and spread. For three dice we get

```
c<-sample(1:6,replace=T,10000)
hist(a+b+c,breaks=2.5:18.5,main="", xlab="three dice")
```

and the bell shape of the normal distribution is starting to emerge. By the time we get to five dice, the **binomial** distribution is virtually indistinguishable from the normal:

```
d<-sample(1:6,replace=T,10000)
e<-sample(1:6,replace=T,10000)
hist(a+b+c+d+e,breaks=4.5:30.5,main="", xlab="five dice")
```

The smooth curve is given by a normal distribution with the same mean and standard deviation:

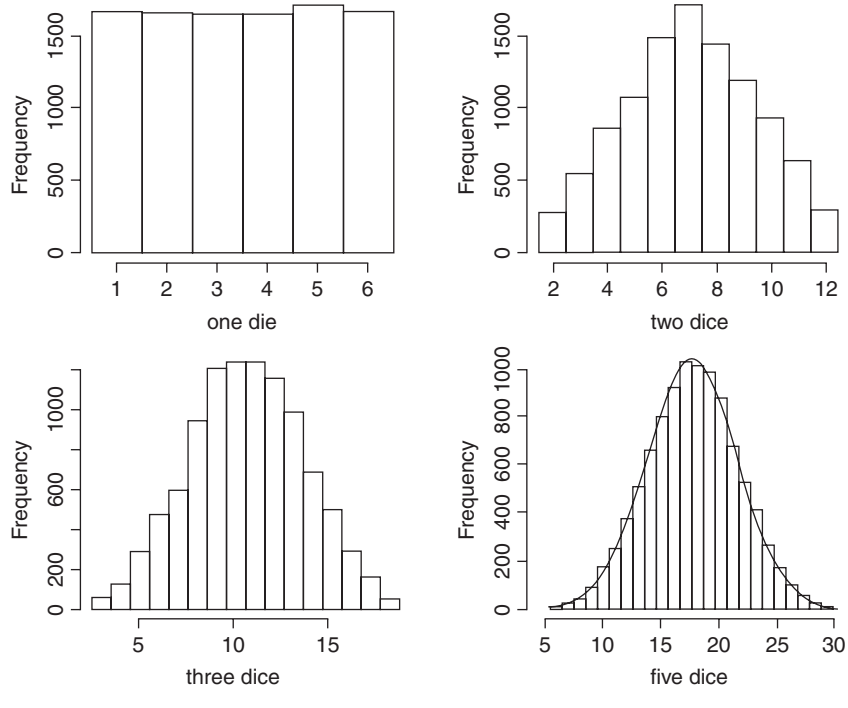
```
mean(a+b+c+d+e)
```

```
[1] 17.5937
```

```
sd(a+b+c+d+e)
```

```
[1] 3.837668
```

```
lines(seq(1,30,0.1),dnorm(seq(1,30,0.1),17.5937,3.837668)*10000)
```



Maximum likelihood with the normal distribution

The probability density of the normal is

$$f(y|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(y-\mu)^2}{2\sigma^2}\right],$$

which is read as saying the probability of getting a data value y, given (|) a mean of μ and a variance of σ^2, is calculated from this rather complicated-looking two-parameter exponential function. For any given combination of μ and σ^2, it gives a value between 0 and 1. Recall that likelihood is the product of the probability densities, for each of the values of the response variable, y. So if we have n values of y in our experiment, the likelihood function is

$$L(\mu, \sigma) = \prod_{i=1}^{n} \left(\frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(y_i - \mu)^2}{2\sigma^2} \right] \right),$$

where the only change is that y has been replaced by y_i and we multiply together the probabilities for each of the n data points. There is a little bit of algebra we can do to simplify this: we can get rid of the product operator, Π, in two steps. First, for the constant term: that, multiplied by itself n times, can just be written as $1/(\sigma\sqrt{2\pi})^n$. Second, remember that the product of a set of antilogs (exp) can be written as the antilog of a sum of the values of x_i like this: $\prod \exp(x_i) = \exp(\sum x_i)$. This means that the product of the right-hand part of the expression can be written as

$$\exp\left[-\frac{\sum_{i=1}^{n} (y_i - \mu)^2}{2\sigma^2} \right]$$

so we can rewrite the likelihood of the normal distribution as

$$L(\mu, \sigma) = \frac{1}{\left(\sigma\sqrt{2\pi}\right)^n} \exp\left[-\frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \mu)^2 \right].$$

The two parameters μ and σ are unknown, and the purpose of the exercise is to use statistical modelling to determine their maximum likelihood values from the data (the n different values of y). So how do we find the values of μ and σ that maximize this likelihood? The answer involves calculus: first we find the derivative of the function with respect to the parameters, then set it to zero, and solve.

It turns out that because of the exp function in the equation, it is easier to work out the log of the likelihood,

$$l(\mu, \sigma) = -\frac{n}{2} \log(2\pi) - n \log(\sigma) - \sum (y_i - \mu)^2 / 2\sigma^2,$$

and maximize this instead. Obviously, the values of the parameters that maximize the log-likelihood $l(\mu, \sigma) = \log(L(\mu, \sigma))$ will be the same as those that maximize the likelihood. From now on, we shall assume that summation is over the index i from 1 to n.

Now for the calculus. We start with the mean, μ. The derivative of the log-likelihood with respect to μ is

$$\frac{\mathrm{d}l}{\mathrm{d}\mu} = \sum (y_i - \mu)/\sigma^2.$$

Set the derivative to zero and solve for μ:

$$\sum (y_i - \mu)/\sigma^2 = 0 \quad \text{so} \quad \sum (y_i - \mu) = 0.$$

Taking the summation through the bracket, and noting that $\sum \mu = n\mu$,

$$\sum y_i - n\mu = 0 \quad \text{so} \quad \sum y_i = n\mu \quad \text{and } \mu = \frac{\sum y_i}{n}.$$

The maximum likelihood estimate of μ is the arithmetic mean.

Next we find the derivative of the log-likelihood with respect to σ:

$$\frac{\mathrm{d}l}{\mathrm{d}\sigma} = -\frac{n}{\sigma} + \frac{\sum (y_i - \mu)^2}{\sigma^3},$$

recalling that the derivative of $\log(x)$ is $1/x$ and the derivative of $-1/x^2$ is $2/x^3$. Solving, we get

$$-\frac{n}{\sigma} + \frac{\sum (y_i - \mu)^2}{\sigma^3} = 0 \quad \text{so} \quad \sum (y_i - \mu)^2 = \sigma^3 \left(\frac{n}{\sigma}\right) = \sigma^2 n$$

$$\sigma^2 = \frac{\sum (y_i - \mu)^2}{n}.$$

The maximum likelihood estimate of the variance σ^2 is the mean squared deviation of the y values from the mean. This is a biased estimate of the variance, however, because it does not take account of the fact that we estimated the value of μ from the data. To unbias the estimate, we need to lose 1 degree of freedom to reflect this fact, and divide the sum of squares by $n - 1$ rather than by n (see p. 52 and restricted maximum likelihood estimators in Chapter 19).

Here, we illustrate R's built-in probability functions in the context of the normal distribution. The density function dnorm has a value of z (a quantile) as its argument. Optional arguments specify the mean and standard deviation (default is the standard normal with mean 0 and standard deviation 1). Values of z outside the range -3.5 to $+3.5$ are very unlikely.

```
par(mfrow=c(2,2))
curve(dnorm,-3,3,xlab="z",ylab="Probability density",main="Density")
```

The probability function pnorm also has a value of z (a quantile) as its argument. Optional arguments specify the mean and standard deviation (default is the standard normal with mean 0 and standard deviation 1). It shows the cumulative probability of a value of z less than or equal to the value specified, and is an S-shaped curve:

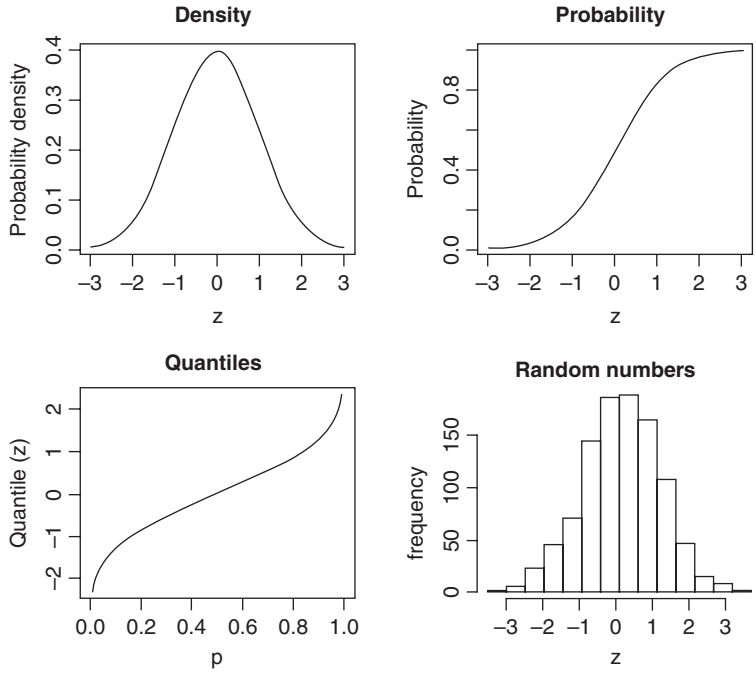
```
curve(pnorm,-3,3,xlab="z",ylab="Probability",main="Probability")
```

Quantiles of the normal distribution qnorm have a cumulative probability as their argument. They perform the opposite function of pnorm, returning a value of z when provided with a probability.

```
curve(qnorm,0,1,xlab="p",ylab="Quantile (z)",main="Quantiles")
```

The normal distribution random number generator rnorm produces random real numbers from a distribution with specified mean and standard deviation. The first argument is the number of numbers that you want to be generated: here are 1000 random numbers with mean 0 and standard deviation 1:

```
y<-rnorm(1000)
hist(y,xlab="z",ylab="frequency",main="Random numbers")
```



The four functions (d, p, q and r) work in similar ways with all the other probability distributions.

Generating random numbers with exact mean standard deviation

If you use a random number generator like rnorm then, naturally, the sample you generate will not have exactly the mean and standard deviation that you specify, and two runs will produce vectors with different means and standard deviations. Suppose we want 100 normal random numbers with a mean of exactly 24 and a standard deviation of precisely 4:

```
yvals<-rnorm(100,24,4)
mean(yvals)
```

```
[1] 24.2958
```

```
sd(yvals)
```

```
[1] 3.5725
```

Close, but not spot on. If you want to generate random numbers with an exact mean and standard deviation, then do the following:

```
ydevs<-rnorm(100,0,1)
```

Now compensate for the fact that the mean is not exactly 0 and the standard deviation is not exactly 1 by expressing all the values as departures from the sample mean scaled in units of the sample's standard deviations:

```
ydevs<-(ydevs-mean(ydevs))/sd(ydevs)
```

Check that the mean is zero and the standard deviation is exactly 1:

```
mean(ydevs)
```

```
[1] -2.449430e-17
```

```
sd(ydevs)
```

```
[1]  1
```

The mean is as close to zero as makes no difference, and the standard deviation is one. Now multiply this vector by your desired standard deviation and add to your desired mean value to get a sample with exactly the means and standard deviation required:

```
yvals<-24 + ydevs*4
mean(yvals)
```

```
[1] 24
```

```
sd(yvals)
```

```
[1] 4
```

Comparing data with a normal distribution

Various tests for normality are described on p. 281. Here we are concerned with the task of comparing a histogram of real data with a smooth normal distribution with the same mean and standard deviation, in order to look for evidence of non-normality (e.g. skew or kurtosis).

```
par(mfrow=c(1,1))
fishes<-read.table("c:\\temp\\fishes.txt",header=T)
attach(fishes)
names(fishes)
```

```
[1] "mass"
```

```
mean(mass)
```

```
[1] 4.194275
```

```
max(mass)
```

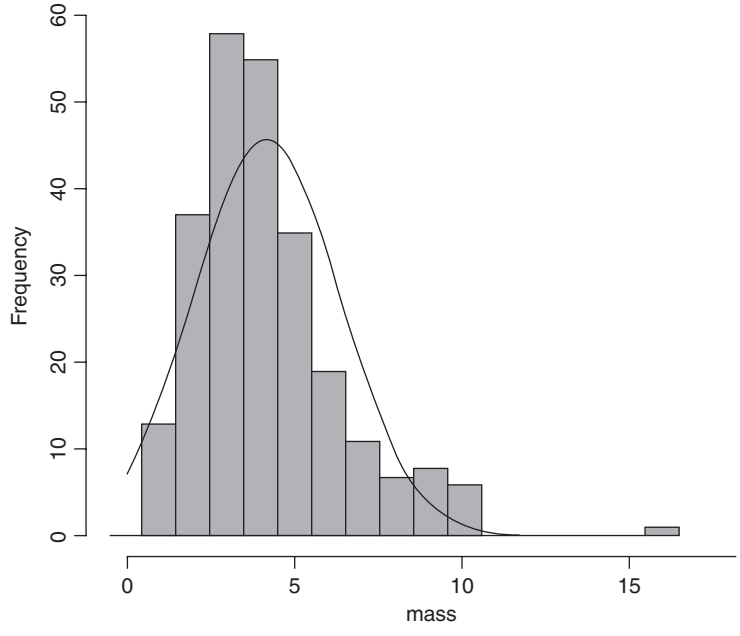
```
[1] 15.53216
```

Now the histogram of the mass of the fish is produced, specifying integer bins that are 1 gram in width, up to a maximum of 16.5 g:

```
hist(mass,breaks=-0.5:16.5,col="green",main="")
```

For the purposes of demonstration, we generate everything we need *inside* the lines function: the sequence of x values for plotting (0 to 16), and the height of the density function (the number of fish (length(mass)) times the probability density for each member of this sequence, for a normal distribution with mean(mass) and standard deviation sqrt(var(mass)) as its parameters, like this:

```
lines(seq(0,16,0.1),length(mass)*dnorm(seq(0,16,0.1),mean(mass),sqrt(var(mass))))
```



The distribution of fish sizes is clearly *not* normal. There are far too many fishes of 3 and 4 grams, too few of 6 or 7 grams, and too many really big fish (more than 8 grams). This kind of skewed distribution is probably better described by a gamma distribution than a normal distribution (see p. 231).

Other distributions used in hypothesis testing

The main distributions used in hypothesis testing are: the **chi-squared**, for testing hypotheses involving count data; **Fisher's *F***, in analysis of variance (ANOVA) for comparing two variances; and **Student's *t***, in small-sample work for comparing two parameter estimates. These distributions tell us the size of the test statistic that could be expected by chance alone when nothing was happening (i.e. when the null hypothesis was true). Given the rule that a big value of the test statistic tells us that something *is* happening, and hence that the null hypothesis is false, these distributions define what constitutes a big value of the test statistic.

For instance, if we are doing a chi-squared test, and our test statistic is 14.3 on 9 d.f., we need to know whether this is a large value (meaning the null hypothesis is false) or a small value (meaning the null hypothesis is accepted, or at least cannot be rejected). In the old days we would have looked up the value in chi-squared tables. We would have looked in the row labelled 9 (the degrees of freedom row) and the column headed by $\alpha = 0.05$. This is the conventional value for the acceptable probability of committing a Type I error: that is to say we allow a 1 in 20 chance of rejecting the null hypothesis when it is actually true; see p. 317). Nowadays, we just type:

```
1-pchisq(14.3,9)
```

```
[1] 0.1120467
```