5

Single Samples

Suppose we have a single sample. The questions we might want to answer are these.

- What is the mean value?
- Is the mean value significantly different from current expectation or theory?
- What is the level of uncertainty associated with our estimate of the mean value?

In order to be reasonably confident that our inferences are correct, we need to establish some facts about the distribution of the data.

- Are the values normally distributed or not?
- Are there outliers in the data?
- If data were collected over a period of time, is there evidence for serial correlation?

Non-normality, outliers and serial correlation can all invalidate inferences made by standard parametric tests like Student's *t*-test. Much better in cases with non-normality and/or outliers to use a non-parametric technique like Wilcoxon's signed-rank test. If there is serial correlation in the data, then you need to use time series analysis or mixed effects models.

Data Summary in the One Sample Case

To see what is involved, read the data called *y* from the file called das.txt

```
data<-read.table("c:\\temp\\das.txt",header=T)
names(data)
```

```
[1] "y"
```

```
attach(data)
```

Statistics: An Introduction using R M. J. Crawley
© 2005 John Wiley & Sons, Ltd ISBNs: 0-470-02298-1 (PBK); 0-470-02297-3 (PPC)

Summarizing the data could not be simpler. We use the built-in function called summary like this:

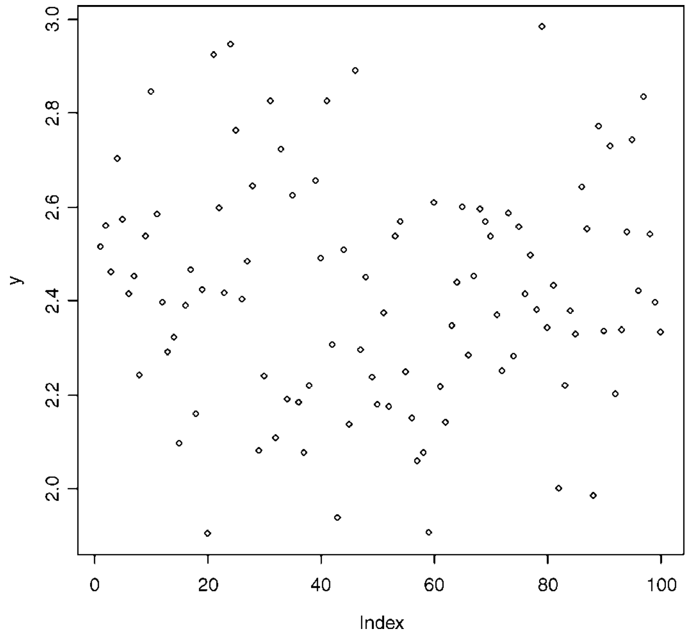
```
summary(y)
```

```
 Min.   1st Qu.   Median    Mean    3rd Qu.   Max.
1.904    2.241    2.414    2.419     2.568   2.984
```

This gives us six pieces of information about the vector called *y*. The smallest value is 1.904 (labelled Min. for minimum) and the largest value is 2.984 (labelled Max. for maximum). There are two measures of central tendency: the median is 2.414 and the arithmetic mean in 2.419. What you may be unfamiliar with are the figures labelled '1st Qu.' and '3rd Qu.'. The 'Qu.' is an abbreviation of quartile, which means one quarter of the data: the first quartile is the value of the data, below which lie the smallest 25% of the data. The median is the 2nd quartile by definition (half the data are smaller than the median). The 3rd quartile is the value of the data, above which lie the largest 25% of the data (it is sometimes called the 75th Percentile, because 75% of the values of *y* are smaller than this value).

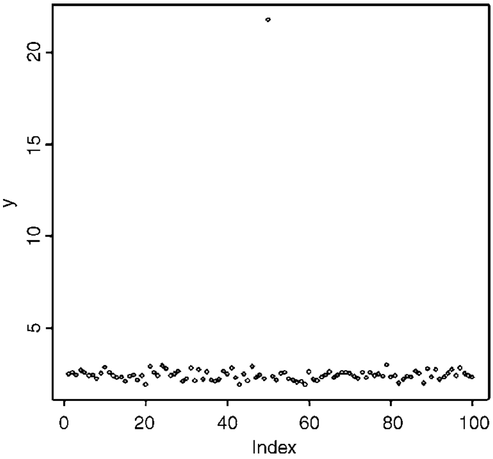
Plotting the data requires us to say exactly what sort of plot we want. If we just say

```
plot(y)
```



we see the values of *y* on the *y* axis, plotted against something called Index. Note that we did not supply the value of Index. Whenever you say plot and there is only one variable, R assumes that you want to plot the values of *y* in the sequence in which they appear in the dataframe, i.e. starting with the first value on the left, in sequence up to the value in position number = length(y) on the right. This is very useful for data checking to make sure that no really silly values appear in *y* (e.g. typing mistakes on data entry). In the present case, suppose the middle value, y[50] had been typed in as 21.79386 instead of 2.179386. Then plot(y) would draw attention to the mistake at once if you write:

```
y[50] < -21.79386
plot(y)
```



The mistake sticks out like a sore thumb, and the error can be rectified as follows. It is not obvious which of the *y* values is wrong, but it is clear that it is the only value bigger than 10. So to find which value it is, we use the which function like this:

```
which(y > 10)
[ 1] 50
```

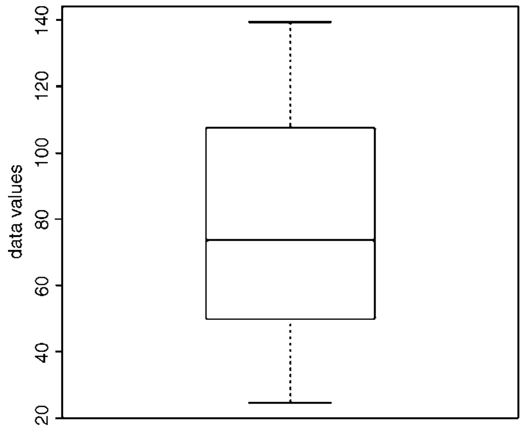
So now we can retype the correct value for the 50th element of *y*:

```
y[50] < - 2.179386
```

and the data are now edited. You could plot them again, to check.

A second kind of plot useful in data summary is the ‘box and whisker plot’. It is a visual representation of the data shown in the summary function, above.

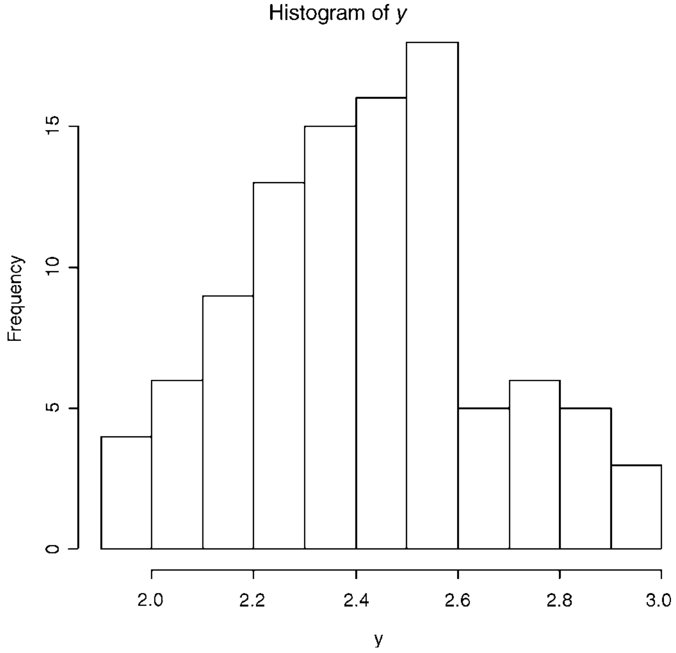
```
boxplot(y,ylab = "data values")
```



The horizontal bar in the middle shows the median value of *y*. The top of the box above the median shows the 75th percentile, and the bottom of the box below the median shows the 25th percentile. Both boxes together show where the middle 50% of the data lie (this is called 'the interquartile range'). The whiskers show the maximum and minimum values of *y* (later on we shall see what happens when the data contain 'outliers').

The last sort of plot that we might want to use for a single sample is the histogram

```
hist(y)
```



This is a very informative plot because it shows that the left-hand side of the distribution is a different shape from the right-hand side. The histogram is said to be ‘skew to the left’ or negatively skew, because there is a longer ‘tail’ to the left of the mode (where there are six bars) than there is to the right (only four bars).

Simple as they seem, there are actually lots of issues about histograms. Perhaps the most important issue is where, exactly, to draw the lines between the bars (the ‘bin widths’ in the jargon). For whole number (integer) data this is often an easy decision (draw a bar of the histogram for each of the integer values of *y*). However, for continuous (real number) data like we have here, that approach is completely inappropriate. How many different values of *y* do we have in our vector of 100 numbers? The appropriate function to answer questions like this is table: we don’t want to see all the values of *y*, we just want to know how many different values of *y* there are. That is to say, we want to know the length of the table of different *y* values:

```
length(table(y))
```

```
[1] 100
```

So there are no repeats of any of the *y* values, and our histogram would be completely uninformative. Let’s look more closely to see what R has chosen on our behalf in designing the histogram. The *x* axis is labelled every 0.2 units, in each of which there are two bars. So the chosen bin width is 0.1. R uses simple rules to select what it thinks will make a ‘pretty’ histogram. It wants to have a reasonable number of bars (too few bars looks dumpy, while too many makes the shape too rough); there are 11 bars in this case. The next criterion is to have ‘sensible’ widths for the bins. It makes more sense, for instance to have the bins 0.1 units wide (as here) than to use one tenth of the range of *y* values, or one eleventh of the range (note the use of the diff and range functions):

```
(max(y)-min(y))/10
```

```
[1] 0.1080075
```

```
diff(range(y))/11
```

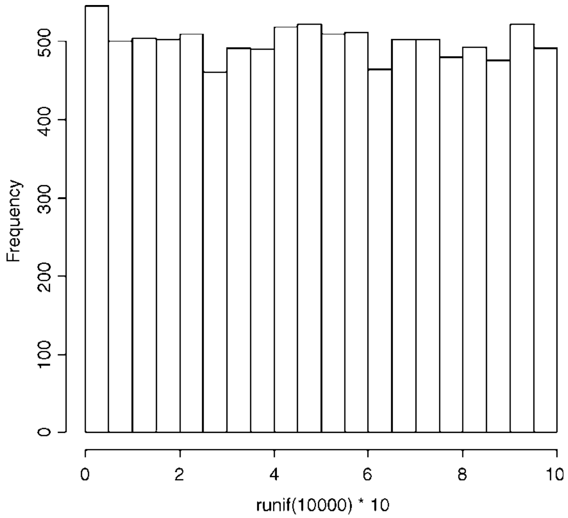
```
[1] 0.09818864
```

So a width of 0.1 is a ‘pretty’ compromise. As we shall see later, you can specify the width of the bins if you don’t like the choice that R has made for you, or if you want to draw two histograms that are exactly comparable.

The Normal Distribution

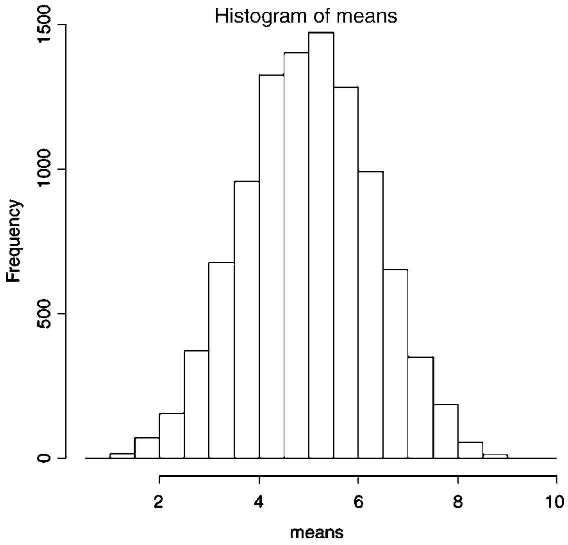
This famous distribution has a central place in statistical analysis. If you take repeated samples from a population and calculate their averages, then these averages will be normally distributed. This is called the **central limit theorem**. Let’s demonstrate it for ourselves. We can take five uniformly 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 five 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)
```

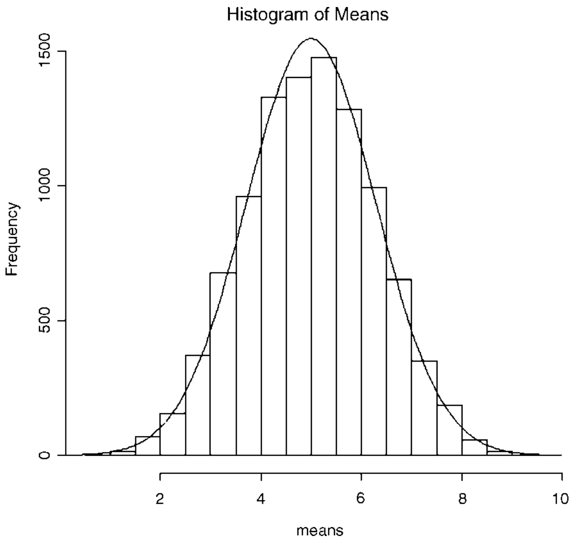
```
[ 1] 1.289960
```

Now we use these two parameters in the p.d.f. (the 'probability density function') of the normal distribution 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 p.d.f. has an integral of 1.0 (that's the area beneath the normal curve), but we had 10 000 samples. Because the normal distribution is symmetrical we therefore expect half of our values to be to the left of the mode; i.e. 10 000 × 0.5 = 5000. To scale the Normal p.d.f. to our particular case, therefore, we multiply by 5000. 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)
```



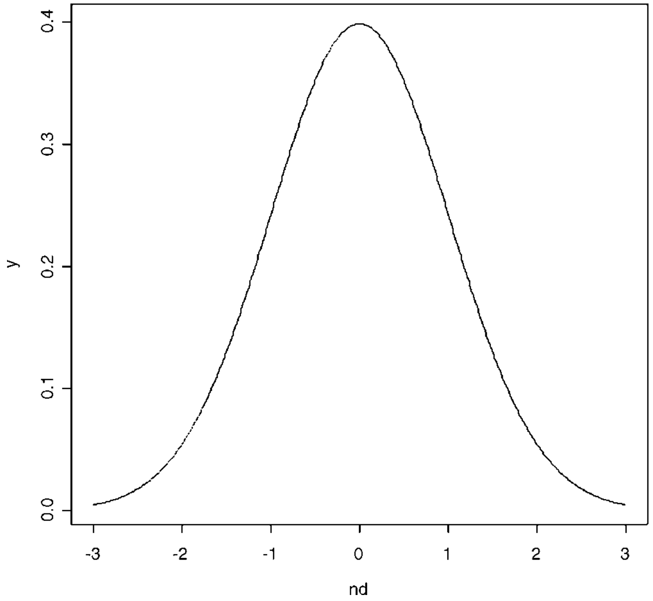
The fit is excellent. The central limit theorem really works. Almost any distribution, even a ‘badly behaved’ one like the negative binomial (p. 242), will produce a Normal distribution of sample means taken from it.

The great thing about the Normal distribution is that we know so much about its shape. Obviously, all values must lie between – infinity and + infinity, so the area under the whole curve is 1.0. The distribution is symmetrical, so half of our samples will fall below the mean, and half will be above it (i.e. the area beneath the curve to the left of the mean is 0.5). The important thing is that we can predict the distribution of samples in various parts of the curve. For example, about 16% of samples will be more than one standard deviation above the mean, and about 2.5% of samples will be more than two standard deviations below the mean; but how do I know this?

There is an infinite number of different possible Normal distributions: the mean can be anything at all, and so can the standard deviation. For convenience, it is useful to have a standard Normal distribution, whose properties we can tabulate. But what would be a sensible choice for the mean of such a standard normal distribution – obviously not 12.7, but what about 1? Not bad, but the distribution is symmetrical, so it would be good to have the left and right halves with similar scales (not 1 to 4 on the right, but −2 to 1 on the left). The only sensible choice is to have the mean = 0. What about the standard deviation? Should that be 0 as well? Hardly, since that would be a distribution with no spread at all. Not very useful. It could be any positive number, but in practice the most sensible choice is 1. So there you have it. The Standard Normal Distribution is one specific case of the Normal with mean = 0 and standard deviation = 1. So how does this help?

It helps a lot, because now we can work out the area below the curve up to any number of standard deviations (these are the values on the *x* axis):

```
nd<-seq(-3,3,0.01)
y<-dnorm(nd)
plot(nd,y,type="l")
```



You can see that almost all values fall within three standard deviations of the mean, one way or the other. It is easy to find the area beneath the curve for any value on the x axis (i.e. for any specified value of the standard deviation). Let's start with s.d. $= -2$. What is the area beneath the curve to the left of -2? It is obviously a small number, but the curvature makes it hard to estimate the area accurately from the plot. R provides the answer with a function called pnorm ('probability for a Normal distribution'; strictly 'cumulative probability' as we shall see). Because we are dealing with a standard Normal (mean $= 0$, s.d. $= 1$) we need only specify the value of the Normal deviate, which is -2 in our case:

```
pnorm(–2)
```

```
[ 1] 0.02275013
```

This tells us that just a bit less than 2.5% of values will be lower than -2. What about one standard deviation below the mean?

```
pnorm(–1)
```

```
[ 1] 0.1586553
```

In this case, about 16% of random samples will be smaller than one standard deviation below the mean. What about big values of the Normal deviate? The histogram shows a maximum of $+3$. What is the probability of getting a sample from a Normal distribution that is more than three standard deviations above the mean? The only point to note here is that pnorm gives the probability of getting a value **less** than the value specified (not more, as we want here). The trick is simply to subtract the value given by pnorm from 1 to get the answer we want:

```
1-pnorm(3)
```

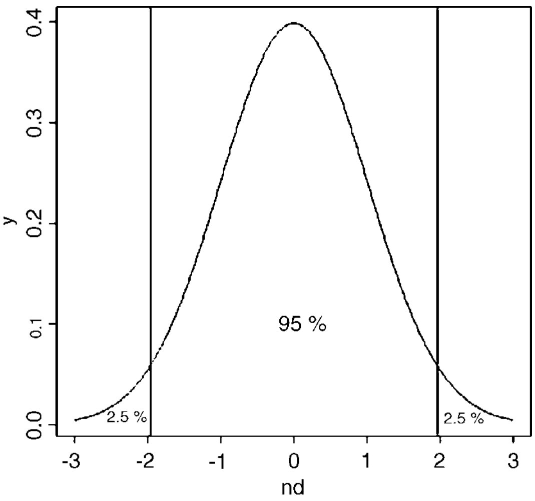
```
[ 1] 0.001349898
```

Such a large value is very unlikely indeed – less than a fifth of 1%, in fact.

Probably the most frequent use of the standard Normal distribution is in working out the values of the Normal deviate that can be expected by chance alone. This, if you like, is the opposite kind of problem to the ones we've just been dealing with. There, we provided a value of the Normal deviate (like -1, or -2 or $+3$) and asked what probability was associated with such a value. Now, we want to provide a probability and find out what value of the Normal deviate is associated with that probability. Let's take an important example. Suppose we want to know the upper and lower values of the Normal deviate between which 95% of samples are expected to lie. This means that 5% of samples will lie outside this range, and because the normal is a symmetrical distribution, this means that 2.5% of values will be expected to be smaller than the lower bound (i.e. lie to the left of the lower bound) and 2.5% of values will be expected to be greater than the upper bound (i.e. lie to the right of the lower bound). The function we need is called qnorm ('quantiles of the Normal distribution') and it is used like this by specifying our two probabilities 0.025 and 0.975 in a vector c(0.025,0.975):

```
qnorm(c(0.025,0.975))
[ 1] -1.959964  1.959964
```

These are two very important numbers in statistics. They tell us that with a Normal distribution, 95% of values will fall between −1.96 and +1.96 standard deviations of the mean. Let's draw these as vertical lines on the normal p.d.f. to see what's involved:



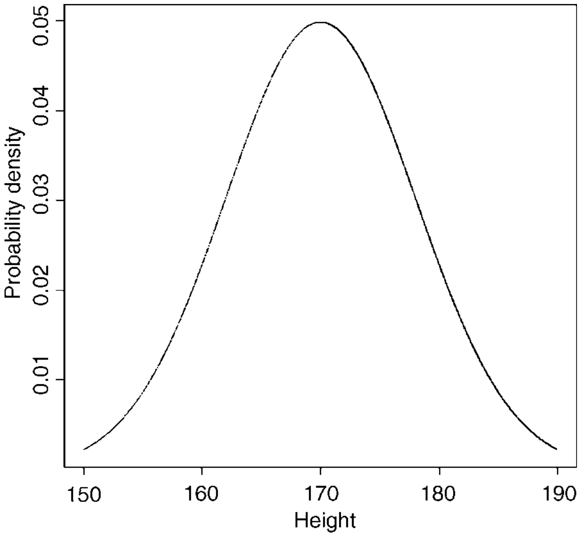
Between the two vertical lines, we can expect 95% of all samples to fall; we expect 2.5% of samples to be less than −1.96 standard deviations below the mean, and we expect 2.5% of samples to be greater than 1.96 standard deviations above the mean. If we discover that this is **not** the case, then our samples are not normally distributed. They might, for instance, follow a Student's *t* distribution (see p. 67).

To sum up, if we want to provide values of the Normal deviate and work out probabilities, we use pnorm; if we want to provide probabilities and work out values of the Normal deviate, we use qnorm. You should try and remember this important distinction.

Calculations using *z* of the Normal Distribution

Suppose we have measured the heights of 100 people. The mean height was 170 cm and the standard deviation was 8 cm. The Normal distribution looks like this:

```
ht<-seq(150,190,0.01)
plot(ht,dnorm(ht,170,8),type="l",ylab="Probability density",xlab="Height")
```



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 zero and a standard deviation of one. 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 of the distribution. 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 is just over 10%. The second question is: what is the probability of selecting one of our people and finding that they are taller than 185 cm? 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 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:

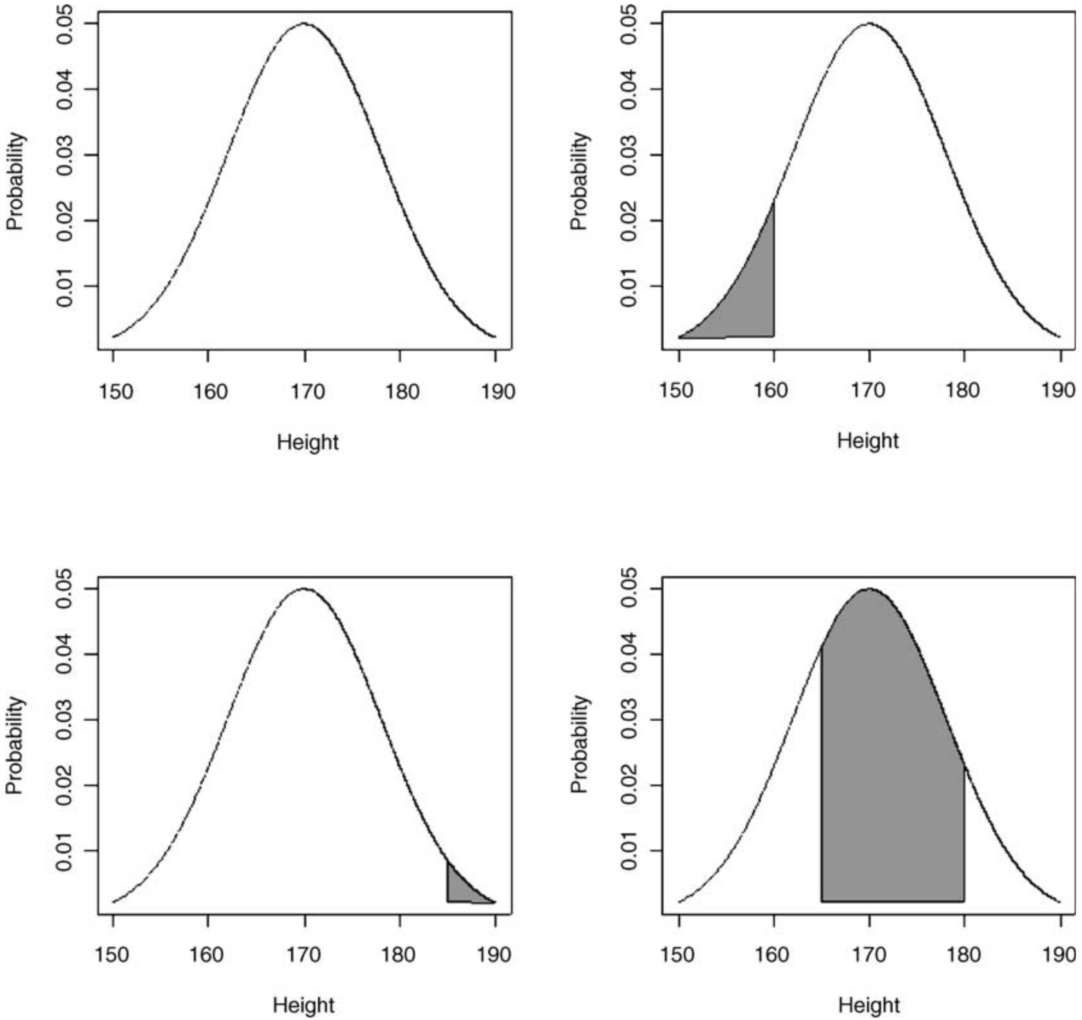
$$z_1 = \frac{(165 - 170)}{8} = -0.625 \text{ and } z_2 = \frac{(180 - 170)}{8} = 1.25.$$

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. It might help to sketch the Normal curve and shade in the area you are interested in:

```
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.


Probability
0.01
0.02
0.03
0.04
0.05
150
160
170
180
190
Height

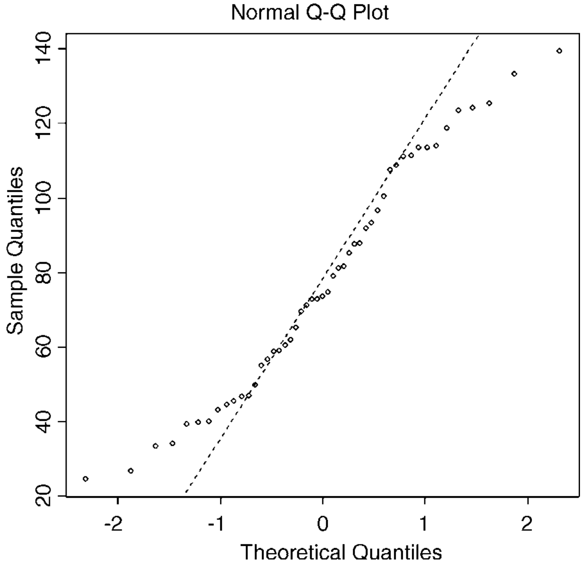
The function called polygon is used for colouring-in different shaped areas under the curve: to see how it is used refer to the figure-generating code on the web site (see Preface), or type

```
?polygon
```

Plots for Testing Normality of Single Samples

The simplest test of normality (and in many ways the best) is the ‘quantile–quantile plot’; it plots the ranked samples from our distribution against a similar number of ranked quantiles taken from a normal distribution. If the sample is normally distributed then the line will be straight. Departures from normality show up as various sorts of non-linearity (e.g. S-shapes, or banana shapes). The functions you need are qqnorm and qqline (quantile–quantile plot against a Normal distribution):

```
qqnorm(y)
qqline(y,lty=2)
```



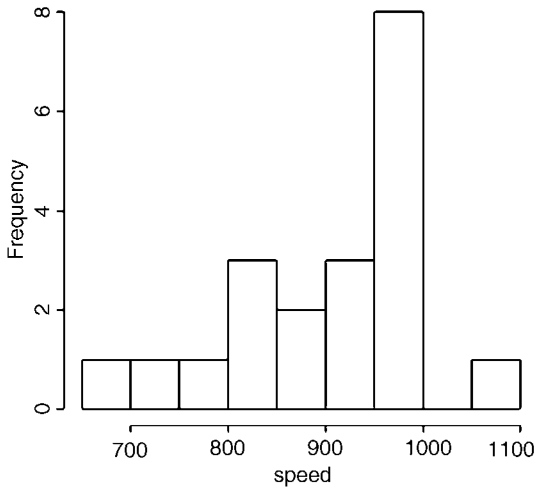
This shows a marked S-shape, indicative of non-normality (as we already know, our distribution is non-Normal because it is skew to the left).

We can investigate the issues involved with Michelson’s (1879) famous data on estimating the speed of light. The actual speed is 299 000 km s^{-1} plus the values in our dataframe called light:

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

```
[ 1] "speed"
```

```
hist(speed)
```



We get a **summary** of the non-parametric descriptors of the sample like this:

```
summary(speed)
```

```
Min.    1st Qu.  Median    Mean   3rd Qu.   Max.
 650        850     940     909       980   1070
```

From this, you see at once that the median (940) is substantially bigger than the mean (909), as a consequence of the strong negative skew in the data seen in the histogram. The **interquartile range** is the difference between the 1st and 3rd quartiles: 980 − 850 = 130. This is useful in the detection of outliers: a good rule of thumb is that an **outlier** is a value more than 1.5 times the interquartile range above the 3rd quartile, or below the 1st quartile $(130 \times 1.5 = 195)$. In this case, therefore, outliers would be measurements of speed that were less than $850 - 195 = 655$ or greater than $980 + 195 = 1175$. You will see that there are no large outliers in this data set, but one or more small outliers (the minimum is 650).

Inference in the One-sample Case

We now know that the speed of light is 299 792.458 km/s. We want to test the hypothesis that Michelson's estimate of the speed of light is significantly different from the value of 299 990 km/s thought to prevail at the time. The data have all had 299 000 subtracted from them, so the test value is 990. Because of the non-Normality, the use of Student's *t*-test in this case is ill advised. The correct test is Wilcoxon's signed rank test. The code for this is in a library of 'Classical Tests' called ctest:

```
library(ctest)
wilcox.test(speed,mu = 990)
```

```
        Wilcoxon signed rank test with continuity correction
data: speed
V = 22.5, p-value = 0.00213
alternative hypothesis: true mu is not equal to 990

Warning message:
Cannot compute exact p-value with ties in: wilcox.
test.default (speed, mu = 990)
```

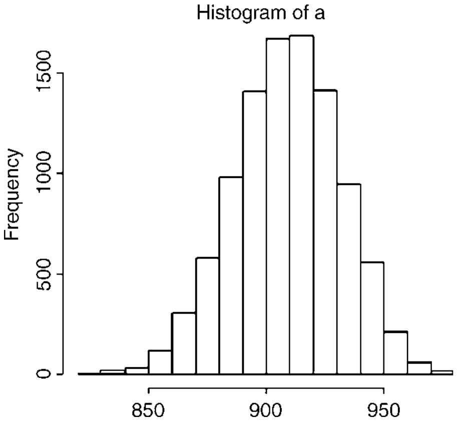
We reject the null hypothesis and accept the alternative hypothesis because $p = 0.00213$ (i.e. much less than 0.05). The speed of light is significantly less than 990.

Bootstrap in Hypothesis Testing with Single Samples

We shall meet parametric methods for hypothesis testing later. Here we use bootstrapping to illustrate another non-parametric method of hypothesis testing. Our sample mean value of y is 909. The question we have been asked to address is this: 'how likely is it that the population mean that we are trying to estimate with our random sample of 100 values is as big as 990?'.

We take 10 000 random samples with replacement using $n = 100$ from the 100 values of light and calculate 10 000 values of the mean. Then we ask: what is the probability of obtaining a mean as large as 990 by inspecting the right-hand tail of the cumulative probability distribution of our 10 000 bootstrapped mean values? This is not as hard as it sounds:

```
a < -numeric(10000)
for(i in 1:10000) a[i] < -mean(sample(speed,replace = T))
hist(a)
```



The test value of 990 is off the scale to the right. A mean of 990 is clearly most unlikely, given the data:

```
max(a)
```

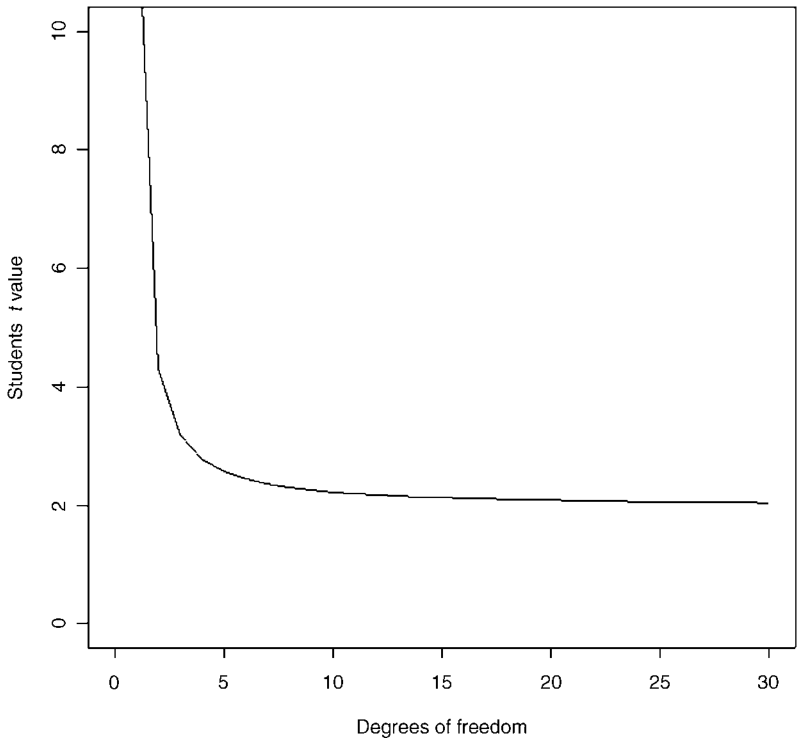
```
[ 1] 979
```

In our 10 000 samples of the data, we never obtained a mean value greater than 979, so the probability that the mean is 990 is clearly $p < 0.0001$.

Student's *t*-distribution

Student's *t*-distribution is used instead of the Normal distribution when sample sizes are small ($n < 30$). Remember that the 95% intervals of the standard normal were -1.96 to $+1.96$ standard deviations. Student's *t*-distribution produces bigger intervals than this. The smaller the sample, the bigger the interval. Let's see this in action. The equivalents of pnorm and qnorm are pt and qt. We are going to plot a graph to show how the upper interval (equivalent to the Normal's 1.96) varies with sample size in a *t*-distribution. This is a deviate so the appropriate function is qt. We need to supply it with the probability (in this case $p = 0.975$) and the degrees of freedom (we'll vary these from 1 to 30 to produce the graph)

```
plot(c(0,30),c(0,10),type="n",xlab="Degrees of freedom",ylab= "Students t value")
lines(1:30,qt(0.975,df=1:30))
```



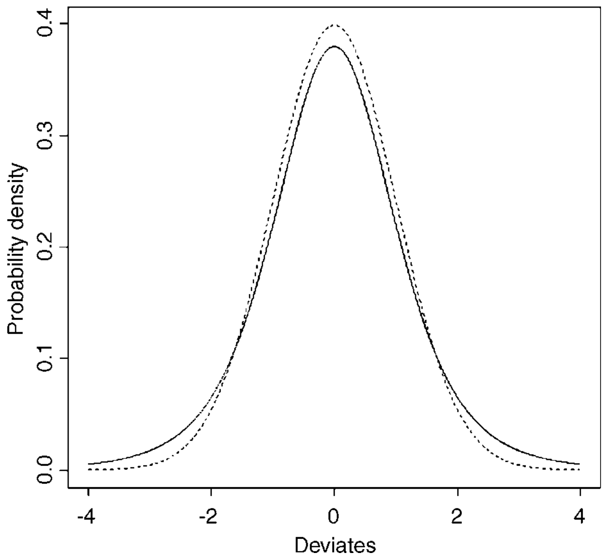
The importance of using Student's *t* rather than the Normal is relatively slight until the degrees of freedom fall below about ten (above which the critical value is roughly two), and then it increases dramatically below about five degrees of freedom. For samples with more than 30 degrees of freedom, Student's *t* produces an asymptotic value of 1.96, just like the Normal. This graph demonstrates that Student's $t = 2$ is a reasonable rule of thumb; memorizing this will save you lots of time in looking up critical values in later life.

So what does the *t*-distribution look like, compared to a Normal distribution? Let's redraw the standard normal as a dotted line (lty = 2):

```
xvs < -seq(-4,4,0.01)
plot(xvs,dnorm(xvs),type = "l",lty = 2,ylab = "Probability density", xlab = "Deviates")
```

Now we can overlay Student's *t* with d.f. = 5 as a solid line to see the difference:

```
lines(xvs,dt(xvs,df = 5))
```



The difference between the Normal (dotted) and Student's *t*-distributions (solid line) is that the *t*-distribution has 'fatter tails'. This means that extreme values are more likely with a *t*-distribution than with a Normal, and the confidence intervals are correspondingly broader. So instead of a 95% interval of ±1.96 with a Normal distribution we should have a 95% interval of ±2.57 for a Student's *t*-distribution with five degrees of freedom:

```
qt(0.975,5)
[ 1] 2.570582
```

Higher-order Moments of a Distribution

So far, and without saying so explicitly, we have encountered the first two moments of a sample distribution. The quantity $\sum y$ was used in the context of defining the arithmetic mean of a single sample: this is the first moment $\bar{y} = \sum y/n$. The quantity $\sum (y - \bar{y})^2$, the sum of squares, was used in calculating sample variance, and this is the second moment of the distribution, $s^2 = \sum (y - \bar{y})^2/(n - 1)$. Higher-order moments involve powers of the difference greater than two like $\sum (y - \bar{y})^3$ and $\sum (y - \bar{y})^4$.

Skew

Skew (or skewness) is the dimensionless version of the third moment about the mean

$$m_3 = \frac{\sum (y - \bar{y})^3}{n}$$

which is rendered dimensionless by dividing by the cube of the standard deviation of y (because this is also measured in units of y^3):

$$s_3 = \text{s.d.}\,(y)^3 = (\sqrt{s^2})^3.$$

The skew is then given by

$$\text{skew} = \gamma_1 = \frac{m_3}{s_3}.$$

It measures the extent to which a distribution has long, drawn out **tails** on one side or the other. A Normal distribution is symmetrical and has skew = 0. Negative values of γ_1 mean skew to the left (negative skew) and positive values mean skew to the right. To test whether a particular value of skew is significantly different from 0 (and hence the distribution from which it was calculated is significantly non-Normal) we divide the estimate of skew by its approximate standard error:

$$\text{s.e.}_{\gamma_1} = \sqrt{\frac{6}{n}}.$$

It is straightforward to write an R function to calculate the degree of skew for any vector of numbers, x, like this:

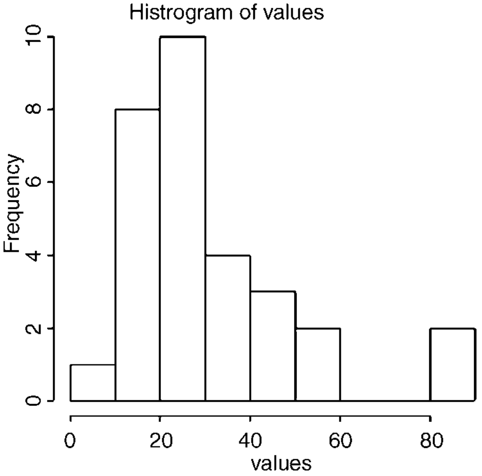
```
skew < -function(x){
m3 < -sum((x-mean(x))^3)/length(x)
s3 < -sqrt(var(x))^3
m3/s3 }
```

Note the use of the length(x) function to work out the sample size, n, whatever the size of the vector x. The last expression inside a function is not assigned to a variable name, and is returned as the value of skew(x) when this is executed from the command line.

```
data < -read.table("c:\\temp\\skewdata.txt",header = T)
attach(data)
names(data)
```

```
[ 1] "values"
```

```
hist(values)
```



The data appear to be positively skewed (i.e. to have a longer tail on the right than on the left). We use the new function skew to quantify the degree of skewness:

```
skew(values)
```

```
[ 1] 1.318905
```

Now we need to know whether a skew of 1.319 is significantly different from zero. We do a t-test, dividing the observed value of skew by its standard error $\sqrt{6/n}$

```
skew(values)/sqrt(6/length(values))
```

```
[ 1] 2.949161
```

Finally we ask, what is the probability of getting a t-value of 2.949 by chance alone, when the skew value really is zero?

```
1-pt(2.949,28)
```

```
[ 1] 0.003185136
```

We conclude that these data show significant non-Normality ($p < 0.0032$).

The next step might be to look for a transformation that normalizes the data by reducing the skewness. One way of drawing in the larger values is to take square roots, so let's try this to begin with:

```
skew(sqrt(values))/sqrt(6/length(values))
```

```
[ 1] 1.474851
```

This is not significantly skew. Alternatively, we might take the logs of the values:

```
skew(log(values))/sqrt(6/length(values))
```

```
[ 1] -0.6600605
```

This is now slightly skew to the left (negative skew), but the value of Student's t is smaller than with a square root transformation, so we might prefer a log transformation in this case.

Kurtosis

This is a measure of non-Normality that has to do with the peakyness, or flat-toppedness, of a distribution. The Normal distribution is bell shaped, whereas a kurtotic distribution is other than bell shaped. In particular, a more flat-topped distribution is said to be platykurtotic, and a more pointy distribution is said to be leptokurtotic. Kurtosis is the dimensionless version of the fourth moment about the mean

$$m_4 = \frac{\sum (y - \bar{y})^4}{n},$$

which is rendered dimensionless by dividing by the square of the variance of y (because this is also measured in units of y^4):

$$s_4 = \mathrm{var}(y)^2 = (s^2)^2.$$

Kurtosis is then given by

$$\text{kurtosis} = \gamma_2 = \frac{m_4}{s_4}\text{-}3.$$

The minus 3 is included because a Normal distribution has $m_4/s_4 = 3$. This formulation therefore has the desirable property of giving zero kurtosis for a Normal distribution, while a flat-topped (platykurtic) distribution has a negative value of kurtosis, and a pointy (leptokurtic) distribution has a positive value of kurtosis. The approximate standard error of kurtosis is

$$\text{s.e.}_{\gamma_2} = \sqrt{\frac{24}{n}}.$$

An R function to calculate kurtosis might look like this:

```
kurtosis<-function(x) {
m4<-sum((x-mean(x))^4)/length(x)
s4<-var(x)^2
m4/s4 - 3 }
```

For our present data, we find that kurtosis is not significantly different from Normal:

```
kurtosis(values)
[1] 1.297751
kurtosis(values)/sqrt(24/length(values))
[1] 1.450930
```