STATISTICS: AN INTRODUCTION USING R

By M.J. Crawley

Exercises

3. STATISTICS OF ONE AND TWO SAMPLES

The hardest part of any statistical work is knowing how to get started. One of the hardest things about getting started in choosing the right kind of statistical analysis for your data and your particular scientific question. The truth is that there is no substitute for experience. The way to know what to do is to have done it properly lots of times before. But here are some useful guidelines. It is essential, for example to know:

1) Which of your variables is the response variable?

2) Which are the explanatory variables?

3) Are the explanatory variables continuous or categorical, or a mixture of both?

4) What kind of response variable have you got: is it a continuous measurement, a count, a proportion, a time-at-death or a category ?

The answers to these questions should lead you quickly to the appropriate choice if you use the following dichotomous key. It begins by asking questions about the nature of your explanatory variables, and ends by asking about what kind of response variable you have got.

1. Explanatory variables all categorical		**2**
At least one explanatory variable a continuous measurement		**4**
2. Response variable a count	**Contingency table**	
Response variable not a count		**3**
3. Response variable a continuous measurement	**Analysis of variance**	
Response variable other than this	**Analysis of deviance**	
4. All explanatory variables continuous	**Regression**	**5**
Explanatory variables both continuous and categorical	**Analysis of covariance**	**5**
5. Response variable continuous	**Regression** or **Ancova**	
Response variable a count	**Log-linear** models (Poisson errors)	
Response variable a proportion	**Logistic** model (binomial errors)	
Response variable a time-at-death	**Survival analysis**	
Response variable binary	**Binary** logistic analysis	
Explanatory variable is time	**Time series** analysis	

Estimating parameters from data

Data have a number of important properties, and it is useful to be able to quantify the following attributes:

- sample size
- central tendency
- variation
- skew
- kurtosis

The sample size is easy: the number of measurements of the response variable is known universally as *n*. In the words of the old statistical joke: "It's the *n*'s that justify the means". Determining what sample size you *need* in order to address a particular question in specific circumstances is an important question that we deal with later. In R, you find the size of a vector using length(name).

Central tendency

Most data show some propensity to cluster around a central value. Averages from repeated bouts of sampling show a remarkable tendency to cluster around the arithmetic mean (this is the Central Limit Theorem; see below). There are several ways of estimating the central tendency of a set of data and they often differ from one another. These differences can be highly informative about the nature of the data set.

Mode

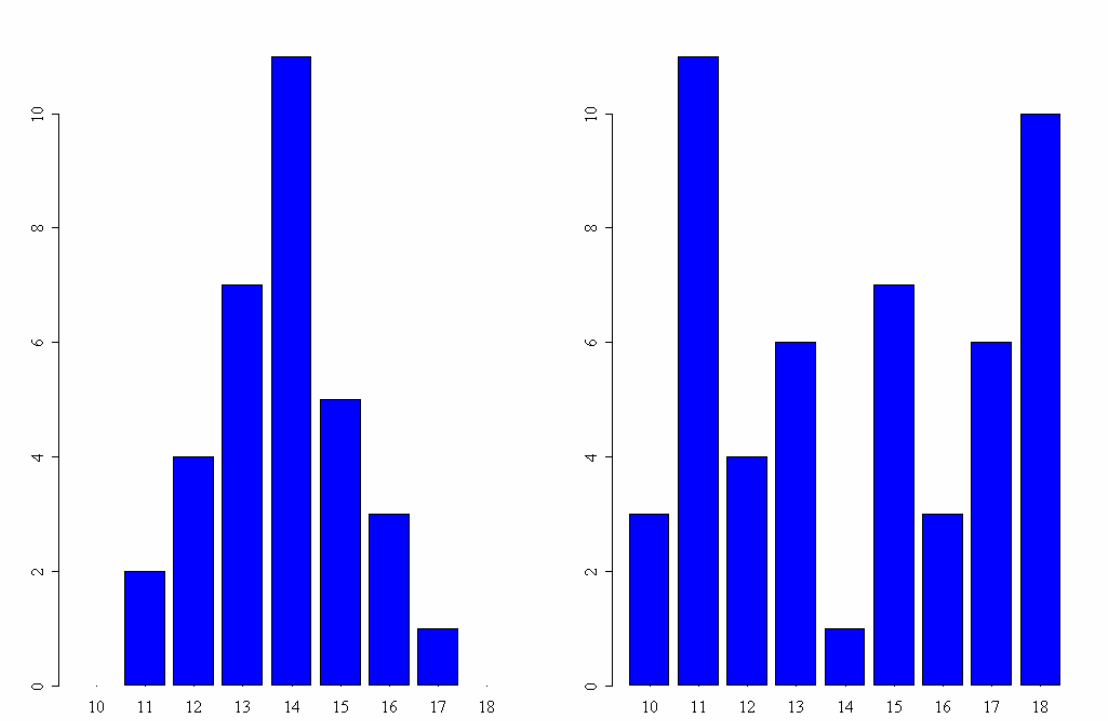
Perhaps the simplest measure of central tendency is the mode: the most frequently represented class of data. Some distributions have several modes, and these can give a poor estimate of central tendency.

```
rm(x)    # this deletes any existing vector called x
distribution<-read.table("c:\\temp\\mode.txt",header=T)
attach(distribution)
names(distribution)
```

```
[1] "fx" "fy"  "fz" "x"
```

To draw the two histograms side by side we set up 2 plotting panels:

```
par(mfrow=c(1,2))
barplot(fx,names=as.character(x))
barplot(fy,names=as.character(x))
```



The mode is an excellent descriptor of the central tendency of the data set on the left, but the two largest modes of the data set on the right are both very poor descriptors of its central tendency.

Median

The median is an extremely appealing measure of central tendency. It is the value of the response variable that lies in the middle of the ranked set of y values. That is to say 50% of the y values are smaller than the median, and 50% are larger. It has the great advantage of not being sensitive to outliers. Suppose we have a vector, y, that contains all of the x values in our left hand histogram. That is to say each *x* value is **rep**eated *fx* times:

```
y<-rep(x,fx)
```

Now to find the median from first principles we **sort** the y values (in this case they are already in ascending order, but we overlook this for the sake of generality).

```
y<-sort(y)
```

Now we need to know how many y values (n) there are; use the **length** directive for this

```
length(y)
```

```
[1] 33
```

Because this is an odd number it means that the median is uniquely determined as the y value in position 17 of the ranked values of y.

```
ceiling(length(y)/2)
```

```
[1] 17
```

We find this value of y using subcripts [].

```
y[17]
```

or, more generally, using the Up Arrow to edit in y[ ] to the **ceiling** directive

```
y[ceiling(length(y)/2)]
```

```
[1] 14
```

Had the vector been of even-numbered length, we could have taken the y values on either side of the mid point (y[length/2] and y[length/2 + 1]) and averaged them. A function to do this is described later). You will not be surprised to learn that there is a built in **median** function, used directly like this:

```
median(y)
```

```
[1] 14
```

Arithmetic mean

The most familiar measure of central tendency is the arithmetic mean. It is the sum of the *y* values divided by the number of *y* values (*n*). Throughout the book we use $\sum$, capital Greek sigma, to mean "add up all the values of". So the mean of *y*, called "y bar", can be written as

$$\bar{y} = \frac{\sum y}{n}$$

So we can compute this using the **sum** and **length** functions

```
sum(y)/length(y)
```

```
[1] 13.78788
```

but you will be not be surprised to learn that there is a built-in function called **mean**

```
mean(y)
```

```
[1] 13.78788
```

Notice that in this case, the arithmetic mean is very close to the median (14.0).

Geometric mean

This is much less familiar than the arithmetic mean, but it has important uses in the calculation of average rates of change in economics and average population sizes in ecology. Its definition appears rather curious at first. It is the n^{th} root of the product of the y values. Just as sigma means "add up", so capital Greek pi, $\prod$, means multiply together. So the geometric mean "y curl", is

$$\tilde{y} = \sqrt[n]{\prod y}$$

Recall that roots are fractional powers, so that the square root of x is $x^{1/2} = x^{0.5}$. So to find the geometric mean of y we need to take the product of all the y values, prod(y), to the power of 1/(length of y), like this:

```
prod(y)^(1/length(y))
```

```
[1] 13.71531
```

As you see, in this case the geometric mean is close to both the arithmetic mean (13.788) and the median (14.0). A different way of calculating the geometric mean is to think about the problem in terms of logarithms. The "log of times is plus", so the log of product of the y values is the sum of the logs of the y values. All the logs used in this book are natural logs (base e = 2.781828) unless stated otherwise (see below). So we could work out the mean of log(y) then calculate its antilog:

```
meanlogy<-sum(log(y))/length(y)
meanlogy
```

```
[1] 2.618513
```

Now you need to remember (or learn!) that the natural antilog function is **exp.** So the antilog of meanlogy is

```
exp(meanlogy)
```

```
[1] 13.71531
```

as obtained earlier by a different route. There is no built-in function for geometric mean, but we can easily make one like this:

```
geometric<-function(x)  exp(sum(log(x))/length(x))
```

log means "log to the base e". Then try the function out using our vector called y:

```
geometric(y)
```

```
[1] 13.71531
```

The reason that the geometric mean is so important can be seen by using a more extreme data set. Suppose we had made the following counts of aphids on different plants:

```
aphid<-c(10,1,1,10,1000)
```

Now the arithmetic mean is hopeless as a descriptor of central tendency:

```
mean(aphid)
```

```
[1] 204.4
```

This measure isn't even close to *any* of the 5 data points. The reason for this, of course, is that the arithmetic mean is so sensitive to outliers, and the single count of 1000 has an enormous influence on the mean.

We can break the rules, and just once use logs to the base 10 to look at this example. The $\log_{10}$ values of the 5 counts are 1, 0, 0, 1 and 3. So the sum of the logs is 5 and the average of the logs is 5/5 = 1. Antilog base 10 of 1 is $10^1 = 10$. This is the geometric mean of these data, and is a much better measure of central tendency (2 of the 5 values are exactly like this) We can check using our own function:

```
geometric(aphid)
```

```
[1] 10
```

Needless to say, we get the same answer, whatever base of logarithms we use. We always use geometric mean to average variables that change multiplicatively, like ecological populations or bank accounts accruing compound interest.

Harmonic mean

Suppose you are working on elephant behaviour. Your focal animal has a square home range with sides of length 2 km. He walks the edge of the home range as follows. The first leg of the journey is carried out at a stately 1 km/hour. He accelerates over the second side to 2 km/hour. He is really into his stride by the 3rd leg of the journey, walking at a dizzying 4 km/hour. Unfortunately, this takes so much out of him that he has to return home at a sluggish 1 km/hour.

The question is this. What is his average speed over the ground. He ended up where he started, so he has no net displacement. But our concern is with his mean velocity. What happens if we work out the average of his 4 speeds?

```
mean(c(1,2,4,1))
```

```
[1] 2
```

This is wrong, as we can see if we work out the average speed from first principles.

$$\text{velocity} = \frac{\text{distance travelled}}{\text{time taken}}$$

The total distance travelled is 4 sides of the square, each of length 2 km, a total of 8 km. Total time taken is a bit more tricky. The first leg took 2 hours (2km at 1 km/hour), the second look 1 hour, the third took half an hour and the last leg took another 2 hours. Total time taken was 2 + 1 + 0.5 + 2 = 5.5 hours. So the correct average velocity is

$$v = \frac{8}{5.5} = 1.455$$

What is the trick? The trick is that this question calls for a different sort of mean. The harmonic mean has a very long definition in words. It is *the reciprocal of the average of the reciprocals.* The reciprocal of x is $\frac{1}{x}$ which can also be written as x^{-1}. So the formula for harmonic mean, "y hat", looks like this:

$$\hat{y} = \frac{1}{\frac{\sum \frac{1}{y}}{n}} = \frac{n}{\sum \frac{1}{y}}$$

Lets try this out with the elephant data:

```
4/sum(1/c(1,2,4,1))
```

```
[1] 1.454545
```

So that works OK. Let's write a general function to compute the harmonic mean of any vector of numbers.

```
harmonic<-function(x) 1/mean(1/x)
```

We can try it out on our vector y:

```
harmonic(y)
```

```
[1] 13.64209
```

To summarise, our measures of central tendency all gave different values, but because the y values were well clustered and there were no serious outliers, all the different values were quite close to one another.

Mode	Median	Arithmetic mean	Geometric mean	Harmonic mean
14	14	13.7879	13.7153	13.6421

Notice that if you try typing mode(y) you don't get 14, you get the message

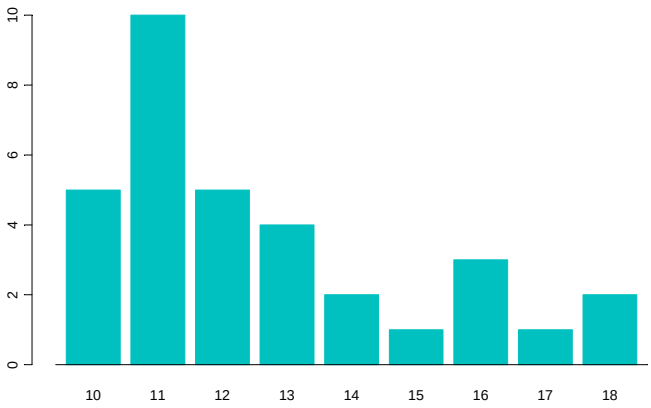
```
mode(y)

[1] "numeric"
```

because **mode** tells you **type** of an S-PLUS object, and y is of type numeric. A different mode is “character”.

The distribution of y was quite symmetric, but many data sets are skew to one side or the other. A long tail to the right is called positive skew, and this is much commoner in practice that data sets which have a long tail to the left (negative skew). The frequency distribution fz is an example of positive skew.

```
par(mfrow=c(1,1))
barplot(fz,names=as.character(x))
```



It is useful to know how the different measures of central tendency compare when the distribution is skew like this. First, we expand the frequency distribution into a vector of measurements, *w*, using **rep** to repeat each of the *x* values *fz* times

```
w<-rep(x,fz)
```

We want to produce a summary of all of the information we have gathered on central tendency for the data in the vector called *w*. For instance,

```
Arithmetic mean  =  12.606
Geometric mean   =  12.404
Harmonic mean    =  12.22
Median           =  12
Modes at 11 and 16 with the largest mode at x = 11
```

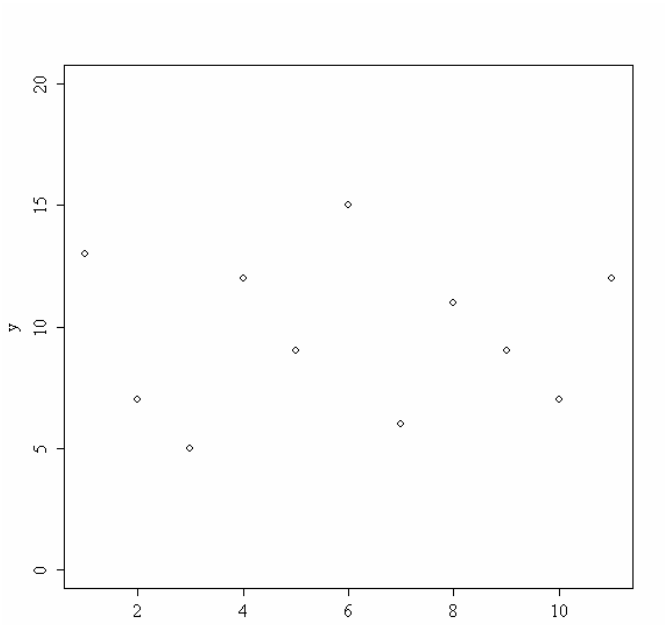
With positive skew, the *mode is the lowest* estimate of central tendency (11.0) and *arithmetic mean is the largest* (12.606). The others are intermediate, with geometric mean > harmonic mean > median in this case.

Measuring variation

A measure of variability is perhaps the most important quantity in statistical analysis. The greater the variability in the data, the greater will be our uncertainty in the values of parameters estimated from the data, and the lower will be our ability to distinguish between competing hypotheses about the data.

Consider the following data, *y*, which are simply plotted in the order in which they were measured:

```
y<-c(13,7,5,12,9,15,6,11,9,7,12)
plot(y,ylim=c(0,20))
```



Visual inspection indicates substantial variation in y. But how to measure it? One way would be to specify the range of *y* values.

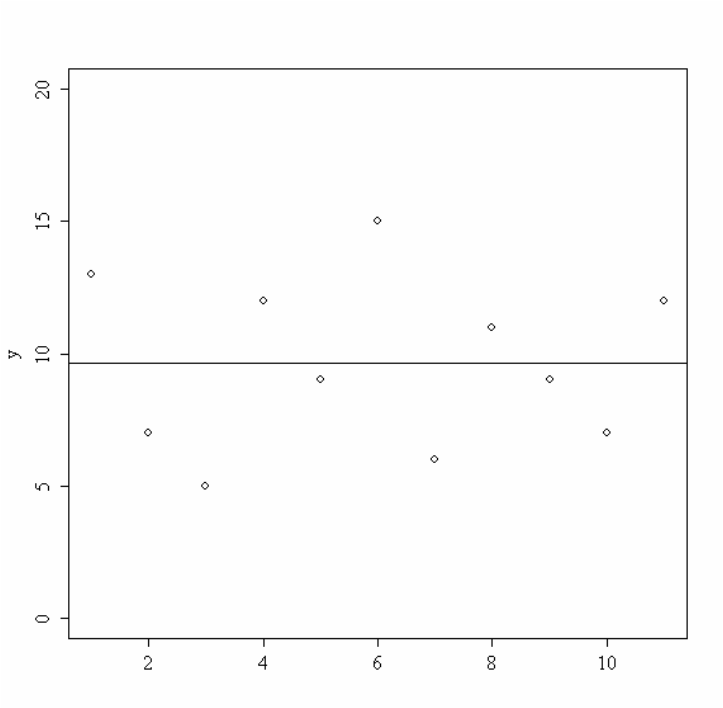
```
range(y)
```

```
[1]  5    15
```

The minimum value of *y* is 5 and the maximum is 15. The variability is contained within the range, and to that extent it is a very useful measure. But it is not ideal for general purposes. For one thing, it is totally determined by outliers, and gives us no indication of more typical levels of variation. Nor is it obvious how to use range in other kinds of calculations (e.g. in uncertainty measures). Finally, the range increases with the sample size, because if you add more numbers then eventually you will add one larger than the current maximum, and if you keep going you will find one smaller than the current minimum. This is a fact, but it is not a property of our unreliability estimate that we particularly want, We would like uncertainty to go down as the sample size went up.

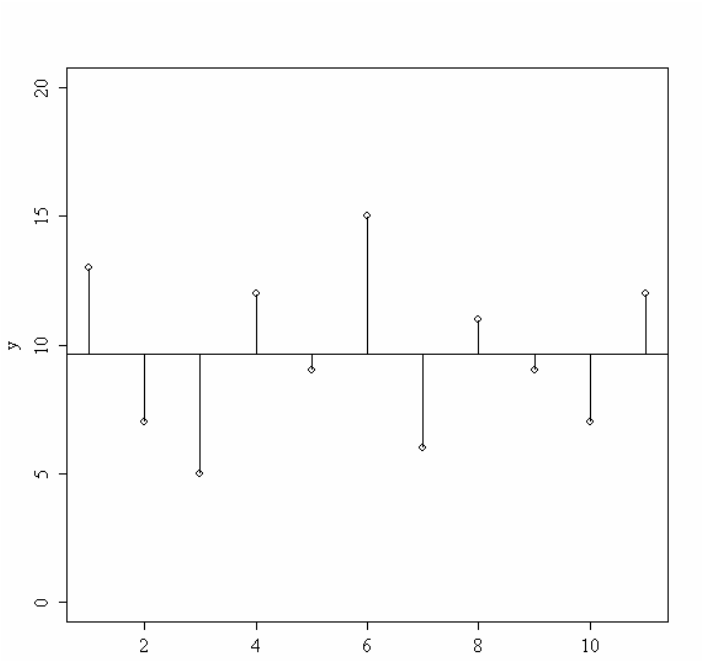
How about fitting the average value of *y* though the data and measuring how far each individual *y* value departs from the mean? The second parameter says the slope of the fitted line is zero:

```
abline(mean(y),0)
```



This divides the data into 5 points that are larger than the mean and 7 points that are smaller than the mean. The distance, *d*, of any point *y* from the mean $\bar{y}$ is

$$d = y - \bar{y}$$



Now a variety of possibilities emerge for measuring variability. How about adding together the different values of *d* ? This turns out to be completely hopeless, because the sum of the *d* values is always zero, no matter what the level of variation in the data !

$$\sum d = \sum (y - \bar{y})$$

Now $\sum \bar{y}$ is the same as $n.\bar{y}$ so

$$\sum d = \sum y - n\bar{y}$$

and we know that $\bar{y} = \sum y / n$ so

$$\sum d = \sum y - \frac{n\sum y}{n}$$

The n's cancel, leaving

$$\sum d \quad = \quad \sum y - \sum y \quad = \quad 0$$

So that's no good then. But wasn't this just a typical mathematician's trick, because the plus and minus values simply cancelled out? Why not ignore the signs and look at the sum of the absolute values of d ?

$$\sum |d| = \sum \left| y - \bar{y} \right|$$

This is actually a very appealing measure of variability because it does not give undue weight to outliers. It was spurned in the past because it made the sums much more difficult, and nobody wants that. It has had a new lease of life since computationally

intensive statistics have become much more popular. The other simple way to get rid of the minus signs is to square each of the *d*'s before adding them up.

$$\sum d^2 = \sum (y - \bar{y})^2$$

This quantity has a fantastically important role in statistics. Given its importance, you might have thought they would have given it a really classy name. Not so. It is called, with appalling literalness the "sum of squares". The sum of squares is the basis of all the measures of variability used in linear statistical analysis.

Is this all we need in our variability measure? Well not quite, because every time we add a new data point to our graph the sum of squares will get bigger. Sometimes by only a small amount when the new value is close to $\bar{y}$ but sometimes by a lot, when it is much bigger or smaller than $\bar{y}$. The solution is straightforward. We calculate the average of the squared deviations (also imaginatively known as the "mean square"). But there is a small hitch. We could not calculate $\sum d^2$ before we knew the value of $\bar{y}$. And how did we know the value of $\bar{y}$. Well we didn't know the value. We estimated it from the data. This leads us into a very important, but simple, concept.

Degrees of freedom

Suppose we had a sample of 5 numbers and their average was 4. What was the sum of the 5 numbers? It must have been 20, otherwise the mean would not have been 4. So now we think about each of the 5 numbers in turn.

We are going to put numbers in each of the 5 boxes. If we allow that the numbers could be positive or negative real numbers we ask how many values could the first number take. Once you see what I'm doing, you will realise it could take any value. Suppose it was a 2.

2				

How many values could the next number take ? It could be anything. Say it was a 7

2	7			

And the 3rd number. Anything. Suppose it was a 4.

2	7	4		

The 4th number could be anything at all. Say it was 0.

2	7	4	0	

Now then. How many values could the last number take? Just 1. It has to be another 7 because the numbers have to add up to 20.

2	7	4	0	7

To recap. We have total freedom in selecting the first number. And the second, third and fourth numbers. But no choice at all in selecting the fifth number. We have 4 degrees of freedom when we have 5 numbers. In general we have n-1 degrees of freedom if we estimated the mean from a sample of size n.

More generally still, we can propose a formal definition of degrees of freedom

Degrees of freedom is the sample size, n, minus the number of parameters, p, estimated from the data.

You should memorise this. In the example we just went through we had $n = 5$ and we had estimated just one parameter from the data: the sample mean, $\bar{y}$. So we had n-1 = 4 d.f.

In a linear regression we estimate two parameters from the data when we fit the model

$$y = a + bx$$

the intercept, *a*, and the slope *b*. Because we have estimated 2 parameters, we have n-2 d.f.. In a one way analysis of variance with 5 genotypes, we estimate 5 means from the data (one for each genotype) so we have n-5 d.f. And so on. The ability to work out degrees of freedom is an incredibly useful skill. It enables you to spot mistakes in experimental designs and in statistical analyses. It helps you to spot pseudoreplication in the work of others, and to avoid it in work of your own.

Variance

We are now in a position to define the most important measure of variability in all of statistics: a variance, s^2, is always

$$\text{variance} = \frac{\text{sum of squares}}{\text{degrees of freedom}}$$

In our case, working out the variance of a single sample, the variance is this:

$$s^2 = \frac{\sum (y - \bar{y})^2}{n - 1}$$

This is so important, we need to take stock at this stage. The data in the following table come from 3 market gardens. The data show the ozone concentrations in parts per hundred million (pphm) on ten summer days.

Garden A	Garden B	Garden C

3	5	3
4	5	3
4	6	2
3	7	1
2	4	10
3	4	4
1	3	3
3	5	11
5	6	3
2	5	10

We want to calculate the variance in ozone concentration for each garden. There are 4 steps to this

- determine the sample mean for each garden
- subtract the sample mean from each value
- square the differences and add them up to get the sum of squares
- divide the sum of squares by degrees of freedom to obtain the variance

We begin by typing the data for each garden into vectors called A, B and C:

```
A<-c(3,4,4,3,2,3,1,3,5,2)

B<-c(5,5,6,7,4,4,3,5,6,5)

C<-c(3,3,2,1,10,4,3,11,3,10)
```

and calculating the 3 sample means

```
mean(A)
[1] 3
mean(B)
[1] 5
mean(C)
[1] 5
```

Now we calculate vectors of length 10 containing the differences $y - \bar{y}$

```
dA <- A-3
dB <- B-5
dC <- C-5
```

then determine the **sum** of the squares of these differences

```
SSA<-sum(dA^2)
SSB<-sum(dB^2)
```

```
SSC<-sum(dC^2)
```

We find that SSA = 12, SSB = 12 and SSC = 128. The 3 variances are obtained by dividing the sum of squares by the degrees of freedom

```
s2A<-SSA/9
s2B<-SSB/9
s2C<-SSC/9
```

To see the values of the 3 variances we type their names separated by semicolons

```
s2A;s2B;s2C
```

```
[1] 1.333333
[1] 1.333333
[1] 14.22222
```

Of course there is a short-cut formula to obtain the sample variance directly

```
s2A<-var(A)
```

There are three important points to be made from this example:

- two populations can have different means but the same variance (Gardens A & B)
- two populations can have the same mean but different variances (Gardens B & C)
- comparing means when the variances are different is an extremely bad idea.

The first two points are straightforward. The third point is profoundly important. Let's look again at the data. Ozone is only damaging to lettuce crops at concentrations in excess of a threshold of 8 pphm. Now looking at the average ozone concentrations we would conclude that both gardens are the same in terms of pollution damage. Wrong ! Look at the data, and count the number of days of damaging air pollution in Garden B. None at all. Now count Garden C. Completely different, with damaging air pollution on 3 days out of 10.

The moral is clear. *If you compare means from populations with different variances you run the risk of making fundamental scientific mistakes.* In this case, concluding there was no pollution damage, when in fact there were damaging levels of pollution 30% of the time.

This begs the question of how you know whether or not 2 variances are significantly different. There is a very simple rule of thumb (the details are explained in Practical 5): if the larger variance is more than 4 times the smaller variance, then the 2 variances are significantly different. In our example here, the variance ratio is

```
14.2222 / 1.3333
[1] 10.66692
```

which is much greater than 4 which means that the variance in Garden C is significantly higher than in the other 2 gardens. Note in passing, that because the variance is the same in Gardens A and B, it is legitimate to carry out a significance test on the difference between their 2 means. This is Student's t-test, explained in full below.

Shortcut formulas for calculating variance

This really was a fairly roundabout sort of process. The main problem with the formula defining variance is that it involves all those subtractions, $y - \bar{y}$. It would be good to find a way of calculating the sum of squares that didn't involve all these subtractions. Let's expand the bracketed term $(y - \bar{y})^2$ to see if we can make any progress towards a subtraction-free solution.

$$(y - \bar{y})^2 \quad = \quad (y - \bar{y})(y - \bar{y}) \quad = \quad y^2 - 2y\bar{y} + \bar{y}^2$$

So far, so good. Now we put the summation through each of the 3 terms separately:

$$\sum y^2 - 2\bar{y}\sum y + n\bar{y}^2 \quad = \quad \sum y^2 - 2\frac{\sum y}{n}\sum y + n\left[\frac{\sum y}{n}\right]^2$$

where only the *y*'s take the summation sign, because we can replace $\sum \bar{y}$ by $n\bar{y}$. We replace $\bar{y}$ with $\sum y / n$ on the r.h.s. Now we cancel the n's and collect the terms

$$\sum y^2 - 2\frac{\left[\sum y\right]^2}{n} + n\frac{\left[\sum y\right]^2}{n^2} \quad = \quad \sum y^2 - \frac{\left[\sum y\right]^2}{n}$$

This gives us a formula for computing the sum of squares that avoids all the tedious subtractions. Actually, it is better computing practice to use the longhand method because it is less subject to rounding errors. Our shortcut formula could be wildly inaccurate if we had to subtract one very large number from another. All we need is the sum of the y's and the sum of the squares of the y's. It is very important to understand the difference between $\sum y^2$ and $\left[\sum y\right]^2$. It is worth doing a numerical example to make plain the distinction between these important quantities.
Let y be {1, 2, 3}. This means that $\sum y^2$ is {1 + 4 + 9} = 14.
The sum of the y's $\sum y$ is {1 + 2 + 3} = 6, so $\left[\sum y\right]^2 = 6^2 = 36$.

The square of the sum is always much larger than the sum of the squares.

Using variance

Variance is used in two main ways

- for establishing measures of unreliability
- for testing hypotheses

Consider the properties that you would like a measure of unreliability to possess. As the variance of the data increases what would happen to unreliability of estimated parameters ? Would it go up or down ? Unreliability would go up as variance increased, so we would want to have the variance on the top of any divisions in our formula for unreliability (i.e. in the numerator).

$$\text{unreliability} \quad \propto \quad s^2$$

What about sample size? Would you want your estimate of unreliability to go up or down as sample size, n, increased ? You would want unreliability to go down as sample size went up, so you would put sample size on the bottom of the formula for unreliability (i.e. in the denominator).

$$\text{unreliability} \quad \propto \quad \frac{s^2}{n}$$

Now consider the units in which unreliability is measured. What are the units in which our current measure are expressed. Sample size is dimensionless, but variance is based on the sum of squared differences, so it has dimensions of mean squared. So if the mean was a length in cm the variance would be an area in cm^2. This is an unfortunate state of affairs. It would make good sense to have the dimensions of the unreliability measure and the parameter whose unreliability it is measuring to be the same. That is why all unreliability measures are enclosed inside a big square root term. Unreliability measures are called *standard errors*. What we have just calculated is the standard error of the mean

$$SE_{\bar{y}} = \sqrt{\frac{s^2}{n}}$$

This is a very important equation and should be memorised. Let's calculate the standard errors of each of our market garden means:

```
sqrt(s2A/10)

[1] 0.3651484

sqrt(s2B/10)

[1] 0.3651484

sqrt(s2C/10)

[1] 1.19257
```

In written work one shows the unreliability of any estimated parameter in a formal, structured way like this.

"The mean ozone concentration in Garden A was 3.0 ± 0.365 (1 s.e., n = 10)"

You write plus or minus, then the unreliability measure then, in brackets, tell the reader what the unreliability measure is (in this case one standard error) and the size of the sample on which the parameter estimate was based (in this case, 10)). This may seem rather stilted, unnecessary even. But the problem is that unless you do this, the reader will not know what kind of unreliability measure you have used. For example, you might have used a 95% confidence interval or a 99% confidence interval instead of 1 s.e..

A confidence interval shows the likely range in which the mean would fall if the sampling exercise were to be repeated. It is a very important concept that people always find difficult to grasp at first. It is pretty clear that the confidence interval will get wider as the unreliability goes up, so

$$\text{confidence interval} \propto \text{unreliability measure} \propto \sqrt{\frac{s^2}{n}}$$

But what do we mean by "confidence" ? This is the hard thing to grasp. Ask yourself this question. Would the interval be wider or narrower if we wanted to be *more* confident that out repeat sample mean falls inside the interval ? It may take some thought, but you should be able to convince yourself that the more confident you want to be, the *wider* the interval will need to be. You can see this clearly by considering the limiting case of complete and absolute certainty. Nothing is certain in statistical science, so the interval would have to be infinitely wide. We can produce confidence intervals of different widths by specifying different levels of confidence. The higher the confidence, the wider the interval.

How exactly does this work. How do we turn the proportionality in the equation above into equality? The answer is by resorting to an appropriate theoretical distribution (see below). Suppose our sample size is too small to use the normal distribution (n < 30, as here), then we traditionally use Student's t distribution. The values of Student's t associated with different levels of confidence are tabulated but also available in the function **qt**, which gives the quantiles of the t distribution. Confidence intervals are always 2-tailed; the parameter may be larger or smaller than our estimate of it. Thus, if we want to establish a 95% confidence interval we need to calculate (or look up) Student's t associated with $\alpha = 0.025$ (i.e. with 0.01*(100 %– 95%)/2). The value is found like this for the left (0.025) and right (0.975) hand tails:

```
qt(.025,9)

[1] -2.262157

qt(.975,9)

[1] 2.262157
```

The first argument is the probability and the second is the degrees of freedom. This says that values as small as -2.262 standard errors below the mean are to be expected in 2.5% of cases (p = 0.025), and values as large as +2.262 standard errors above the mean with similar probability (p = 0.975). *Values of Student's t are* ***numbers of standard errors*** *to be expected with specified probability and for a given number of degrees of freedom*. The values of t for 99% are bigger than these (0.005 in each tail):

```
qt(.995,9)

[1] 3.249836
```

and the value for 99.5% bigger still (0.0025 in each tail):

```
qt(.9975,9)

[1] 3.689662
```

Values of Student's t like these appear in the formula for calculating the width of the confidence interval, and their inclusion is the reason why the width of the confidence interval goes up as our degree of confidence is increased. The other component of the formula, the standard error, is not affected by our choice of confidence level. So, finally, we can write down the formula for the confidence interval of a mean based on a small sample (n < 30):

$$CI_{95\%} = t_{(\alpha=0.025,\ d.f.=9)} \sqrt{\frac{s^2}{n}}$$

For Garden B, therefore, we could write

```
qt(.975,9)*sqrt(1.33333/10)

[1] 0.826022
```

"The mean ozone concentration in Garden B was 5.0 ± 0.826 (95% C.I., n = 10)."

Quantiles

Quantiles are important summary statistics. They show the values of y that are associated with specified percentage points of the distribution of the y values. For instance, the 50% quantile is the same thing as the median. The most commonly used quantiles are aimed at specifying the middle of a data set and the tails of a data set. By the middle of a data set we mean the values of y between which the middle 50% of the numbers lie. That is to say, the values of y that lie between the 25% and 75% quantiles. By the tails of a distribution we mean the extreme values of y: for example, we might define the tails of a distribution as the values that are smaller than the 2.5% quantile or larger than the 97.5% quantile. To see this, we can generate a vector called *z* containing 1000 random numbers drawn from a normal distribution using the

function rnorm, with a mean of 0 and a standard deviation of 1 (the ‘standard normal distribution’ as it is known). This is very straightforward:

```
z<-rnorm(1000)
```

We can see how close the mean really is to 0.0000

```
mean(z)
```

```
[1] -0.01325934
```

Not bad. It is out by just over 1.3%. But what about the tails of the distribution. We know that for an infinitely large sample, the standard normal should have 2.5% of its z values less than –1.96, and 97.5% of its values less than +1.96 (see below). So what is this sample of 1000 points like ? We concatenate the two fractions 0.025 and 0.975 to make the second argument of quantile

```
quantile(z,c(.025,.975))
```

```
      2.5%    97.5%
-1.913038 2.013036
```

Hm. Out by more than 2.5%. Close, but no coconut. It could be just a sample size thing. What if we try with 10,000 numbers ?

```
z<-rnorm(10000)
```

```
quantile(z,c(.025,.975))
```

```
      2.5%    97.5%
-1.985679 1.956595
```

Much better. Clearly the random number generator is a good one, but equally clearly, we should not expect samples of order 1000 to be able to produce exact estimates of the tails of a distribution. This is an important lesson to learn if you intend to use simulation (Monte Carlo methods) in testing statistical models. It says that 10,000 tries is much better than 1,000. Computing time is cheap, so go for 10,000 tries in each run.

Robust estimators

One of the big problems with out standard estimators of central tendency and variability, the mean and the variance, is that they are extremely sensitive to the presence of outliers. We already know how to obtain a robust estimate of central tendency that is not affected by outliers: the median. There are several modern methods of obtaining robust estimators of standard deviation that are less sensitive to outliers. The first is called **mad**: great name, the Median Absolute Deviation. Its value is scaled to be a consistent estimator of the standard deviation from the normal distribution.

```
y<- c(3,4,6,4,5,2,4,5,1,5,4,6)
```

For our existing data set, it looks like this:

```
mad(y)
```

```
[1] 1.4826
```

which is close to, but different from, the standard deviation of y

```
sd(y)
```

```
[1] 1.505042
```

Let's see how the different measures perform in the presence of outliers. We can add an extreme outlier (say y = 100) to our existing data set, using concatenation, and call the new vector y1:

```
y1<-c(y,100)
```

How has the outlier affected the mean (it used to be 4.08333) ?

```
mean(y1)
```

```
[1] 11.46154
```

It has nearly tripled it ! And the standard deviation (it used to be 1.505042) ?

```
sqrt(var(y1))
```

```
[1] 26.64149
```

A more than 17-fold increase. Lets see how the outlier has affected mad's estimate of the standard deviation:

```
mad(y1)
```

```
[1] 1.4826
```

Hardly at all (it was 1.505 before the outlier was added).

These robust estimators suggest a simple function to test for the presence of outliers in a data set. We need to make a decision about the size of the difference between the regular standard deviation and the **mad** in order for us to be circumspect about the influence of outliers. In our extreme example comparing the standard deviations of *y* and *y1* the ratio was 26.64 compared to 1.486 (a nearly 18-fold difference). What if we pick a 4-fold difference as our threshold; it will highlight extreme cases like *y* vs *y1* but it will allow a certain leeway for ecological kinds of data. Let's call the function outlier, and write it like this:

```
outlier<-function(x) {
   if(sqrt(var(x))>4*mad(x))   print("Outliers present")
   else   print("Deviation reasonable") }
```

We can test it by seeing what it makes of the original data set, y:

```
outlier(y)

[1] "Deviation reasonable"
```

What about the data set including y = 100 ?

```
outlier(y1)

[1] "Outliers present"
```

It is good practice to compare robust and standard estimators. I am not suggesting that we do away with means and variances; just that we be aware as to how sensitive they are to extreme values.

Single-sample estimation

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

1) what is the mean value ?

2) is the mean value significantly different from current expectation or theory ?

3) 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.

1) are the values normally distributed or not ?

2) are there outliers in the data ?

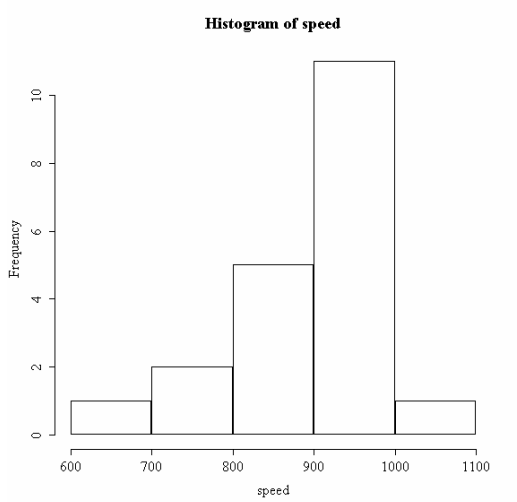
3) if data were collected over a period of time, is there evidence for serial correlation?

Non-normality, outliers and serial correlation can all invalidate inferences made by standard parametric tests like Student's t-test. Much better in such cases to use a robust non-parametric technique like Wilcoxon's signed-rank test.

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 sec^{-1} plus the values in our data frame 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 this

*an **outlier** is a value more than 1.5 times the interquartile range above the 3rd quartile, or below the 1st quartile.*

In this case, 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 1-sample case

We want to test the hypothesis that Michelson's estimate of the speed of light is significantly different from the value of 299,990 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 (note the 'dot')

```
library(ctest)

wilcox.test(speed,mu=990)

Warning: Cannot compute exact p-value with ties
```

```
        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
```

We accept the alternative hypothesis because p = 0.00213 (i.e. much less than 0.05).

For the purpose of demonstration only, we demonstrate the use of Student's t-test

```
t.test(speed,mu=990)

     One-sample t-Test

data:  speed
t = -3.4524, df = 19, p-value = 0.0027
alternative hypothesis: true mean is not equal to 990
95 percent confidence interval:
 859.8931 958.1069
```

This result says that the sample mean is highly significantly different from the null hypothesis value of 990 (p = 0.0027). The 95% confidence interval (859.9 to 958.1) does not include the hypothesised value.

Comparing two means

There are two simple tests for comparing two sample means:

- **Student's t-test** when the means are independent, the variances constant, and the errors are normally distributed

- **Wilcoxon rank sum test** when the means are independent but errors are *not* normally distributed

What to do when these assumptions are violated (e.g. when the variances are different) is discussed later on .

Student was the pseudonym of W.S. Gosset who published his influential paper in *Biometrika* in 1908. He was prevented from publishing under his own name by dint of the archaic employment laws in place at the time which allowed his employer, the Guiness Brewing Company, to prevent him publishing independent work. Student's t distribution, later perfected by R. A. Fisher, revolutionised the study of small sample statistics where inferences need to be made on the basis of the sample variance s^2 with the population variance σ^2 unknown (indeed, usually unknowable). The test statistic is the number of standard errors by which the 2 sample means are separated:

$$t = \frac{\text{difference between the 2 means}}{\text{SE of the difference}} = \frac{\bar{y}_A - \bar{y}_B}{SE_{\text{diff}}}$$

Now we know the standard error of the mean (see p 65) but we have not yet met the standard error of the difference between two means. *The variance of a difference is the sum of the separate variances*. To see this, think about the sum of squares of a difference:

$$\sum\left[(y_A - y_B) - (\mu_A - \mu_B)\right]^2$$

If we average this, we get the variance of the difference (forget about degrees of freedom for a minute). Let's call this average $\sigma^2_{\bar{y}_A - \bar{y}_B}$ and rewrite the sum of squares

$$\sigma^2_{\bar{y}_A - \bar{y}_B} \;=\; \text{average of } [\,(y_A - \mu_A) - (y_B - \mu_B)\,]^2$$

Now square, then expand the brackets, to give

$$(y_A - \mu_A)^2 + (y_B - \mu_B)^2 - 2(y_A - \mu_A)(y_B - \mu_B)$$

We already know that the average of $(y_A - \mu_A)^2$ is the variance of population A and the average of $(y_B - \mu_B)^2$ is the variance of population B. So the variance of the *difference* between the two sample means is the *sum* of the variances of the two samples, plus this term $2(y_A - \mu_A)(y_B - \mu_B)$. But we also know that $\sum d = 0$ (see above) so, because the samples from A and B are independently drawn they are uncorrelated , which means that

$$\text{average of } (y_A - \mu_A)(y_B - \mu_B) \;=\; (y_A - \mu_A)\{\text{average of } (y_B - \mu_B)\} \;=\; 0$$

This important result needs to be stated separately

$$\sigma^2_{\bar{y}_A - \bar{y}_B} \;=\; \sigma^2_A \;+\; \sigma^2_B$$

so if both samples are drawn from populations with the same variance, then *the variance of the difference is twice the variance of an individual mean*.

This allows us to write down the formula for the standard error of the difference between two sample means

$$SE_{\text{difference}} \;=\; \sqrt{\frac{s_A^2}{n_A} + \frac{s_B^2}{n_B}}$$

At this stage we have everything we need to carry out students t-test. Our null hypothesis is that the two sample means are the same, and we shall accept this unless the value of Student's t is so large that it is unlikely that such a difference could have arisen by chance alone. Each sample has 9 degrees of freedom, so we have 18 d.f. in total. Another way of thinking of this is to reason that the complete sample size as 20, and we have estimated 2 parameters from the data, $\bar{y}_A$ and $\bar{y}_B$, so we have 20 – 2 = 18 d.f. We typically use 5% as the chance of rejecting the null hypothesis when it is true

(this is called the Type I error rate). Because we didn't know in advance which of the two gardens was going to have the higher mean ozone concentration (we usually don't), this is a 2-tailed test, so the *critical value* of Student's t is:

```
qt(.975,18)
```

```
[1] 2.100922
```

Thus our test statistic needs to be bigger than 2.1 in order to reject the null hypothesis, and to conclude that the two means are significantly different at $\alpha = 0.05$.

We can write the test like this, using the variances s2A and s2B that we calculated earlier

```
(mean(A)-mean(B))/sqrt(s2A/10+s2B/10)
```

which gives the value of Student's t as

```
[1] -3.872983
```

You won't be at all surprised to learn that there is a built-in function to do all the work for us. It is called, helpfully, **t.test** and is used simply by providing the names of the two vectors containing the samples on which the test is to be carried out (A and B in our case).

```
t.test(A,B)
```

There is rather a lot of output . You often find this. The simpler the statistical test, the more voluminous the output.

```
     Standard Two-Sample t-Test

data:  A and B
t = -3.873, df = 18, p-value = 0.0011
alternative hypothesis: true difference in means is not
equal to 0
95 percent confidence interval:
 -3.0849115 -0.9150885
sample estimates:
 mean of x mean of y
         3         5
```

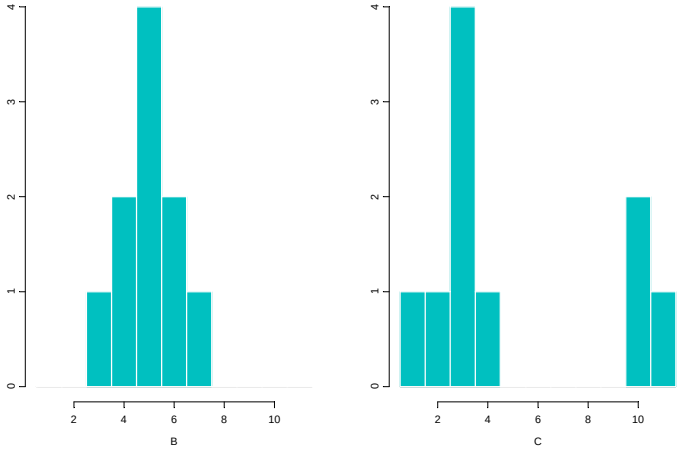
The result is exactly the same as we obtained long hand. The value of t is –3.873 and since *the sign is irrelevant in a t test* we reject the null hypothesis because the test statistic is larger than the critical value of 2.1. The mean ozone concentration is significantly higher in Garden B than in Garden A. The computer print out also gives a p value and a confidence interval. Note that, because the means are significantly different, *the confidence interval on the difference does not include zero* (in fact, it goes from –3.085 up to –0.915). The p value is useful in written work, like this:

“Ozone concentration was significantly higher in Garden B (mean = 5.0 pphm) than in Garden A (mean = 3.0; t = 3.873, p = 0.001, d.f. = 18).”

Wilcoxon rank sum test

A non-parametric alternative to Student’s t test which we could use if the errors looked to be non-normal. Let’s look at the errors in Gardens B and C

```
par(mfrow=c(1,2))
hist(B,breaks=c(0.5:11.5))
hist(C,breaks=c(0.5:11.5))
```



The errors in B look to be reasonably normal, but the errors in C are definitely not normal. The Wilcoxon rank sum test statistic, W, is defined like this. Both samples are put into a single array with their sample names (B and C in this case) clearly attached. Then the aggregate list is sorted, taking care to keep the sample labels with their respective values. Then a rank is assigned to each value, with ties getting appropriate average rank (2-way ties get (rank i + (rank i + 1))/2 , 3-way ties get (rank i + (rank i + 1) + (rank i +2))/3, and so on). Finally the ranks are added up for each of the two samples, and significance is assessed on size of the smaller sum of ranks.

This involves some interesting computing. First we make a combined vector of the samples

```
combined<-c(B,C)
combined
 [1]  5  5  6  7  4  4  3  5  6  5  3  3  2  1 10  4  3
11  3 10
```

then make a list of the sample names, “B” and “C”

```
sample<-c(rep("B",10),rep("C",10))
sample
```

```
 [1] "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "C" "C" "C"
"C" "C" "C" "C" "C" "C" "C"
```

Now the trick is to use the built-in function **rank** to get a vector containing the ranks, smallest to largest, within the combined vector:

```
rank.combi<-rank(combined)
rank.combi
```

```
[1] 12.5 12.5 15.5 17.0  9.0  9.0  5.0 12.5 15.5 12.5  5.0  5.0  2.0  1.0 18.5  9.0  5.0 20.0  5.0 18.5
```

Notice that the ties have been dealt with by averaging the appropriate ranks. Now all we need to do is calculate the sum of the ranks for each garden. We could use **sum** with conditional subscripts for this:

```
sum(rank.combi[sample=="B"])
```

```
[1] 121
```

```
sum(rank.combi[sample=="C"])
```

```
[1] 89
```

Alternatively, we could use **tapply** with **sum** as the required operation

```
tapply(rank.combi,sample,sum)
```

```
   B         C
 121        89
```

In either case, we compare the smaller of the two values (89) with values in Tables (e.g. Snedecor & Cochran, p. 555) and reject the null hypothesis if our value of 89 is *smaller* than the value in tables. For samples of size 10 and 10, the value in tables is 78. Our value is much bigger than this, so we accept the null hypothesis. The two sample means are not significantly different (they are identical, in fact, as we already know).

We can carry out the whole procedure automatically, and avoid the need to use tables of critical values of Wilcoxon's rank sum test, by using the built-in function **wilcox.test**:

```
wilcox.test(B,C)
```

which produces the following output:

```
     Wilcoxon rank-sum test

data:  B and C
rank-sum normal statistic with correction Z = 1.1879, p-
value = 0.2349
alternative hypothesis: true mu is not equal to 0
```

```
Warning messages:
  cannot compute exact p-value with ties in:
wil.rank.sum(x, y, alternative, exact, correct)
```

This is interpreted as follows. The function uses a normal approximation algorithm to work out a z value and from this a p value for the assumption that the means are the same. This p value is much bigger than 0.05, so we accept the null hypothesis. Unhelpfully, it then prints the alternative hypothesis in full, which a careless reader could take as meaning that "true mu is not equal to 0" ("mu" is the difference between the 2 means). We have just demonstrated, rather convincingly, that the true mu *is* equal to 0. The idea of putting the message in the output is to show that we were doing (the default) 2-tailed test, but this is a silly way of saying it, with a serious risk of misinterpretation. The warning message at the end draws attention to the fact that there are ties in the data, and hence the p value can not be calculated exactly (this is seldom a real worry).

Overview

The non-parametric test is much more appropriate than the t-test when the errors are not normal, and the non-parametric is about 95% as powerful with normal errors, and can be *more* powerful than the t-test if the distribution is strongly skewed by the presence of outliers. But the Wilcoxon test does not make the really important point for this case, because like the t-test, it says that ozone pollution in the 2 gardens is not significantly different. Yet we know, because we have looked at the data, that Gardens B and C are definitely different in terms of their ozone pollution, but it is the *variance* that differs, not the mean. Neither the t-test nor the sum rank test can cope properly with situations where the variances are different, but the means are the same.

This draws attention to a very general point:

scientific importance and statistical significance are not the same thing

Lots of results can be highly significant but of no scientific importance at all (e.g. because the effects are cancelled out by later events). Likewise, lost of scientifically important processes may not be statistically significant (e.g. density dependence in population growth rate may not be statistically significant, but leaving it out of a population model because it is not significant will have disastrous effects on the predictions of population dynamics made by the model).

Tests on paired samples

Sometimes, 2-sample data come from paired observations. In this case, we might expect a correlation between the two measurements, either because they were made on the same individual, or were taken from the same location. You might recall that earlier we found that the variance of a difference was the average of

$$(y_A - \mu_A)^2 + (y_B - \mu_B)^2 - 2(y_A - \mu_A)(y_B - \mu_B)$$

which is the variance of sample A plus the variance of sample B minus 2 times the covariance of A and B (see above). When the covariance of A and B is *positive*, this is a great help because it reduces the variance of the difference, and should make it easier to detect significant differences between the means. Pairing is not always effective, because the correlation between y_A and y_B may be weak. It would be disastrous if the correlation were to turn out to be negative!

One way to proceed is to reduce the estimate of the standard error of the difference by taking account of the measured correlation between the two variables. A simpler alternative is to calculate the difference between the two samples from each pair, then do a 1-sample test comparing the mean of the differences to zero. This halves the number of degrees of freedom, but it reduces the error variance substantially if there is a strong positive correlation between y_A and y_B.

The data are a composite biodiversity score based on a kick sample of aquatic invertebrates.

```
x<-c(20,15,10,5,20,15,10,5,20,15,10,5,20,15,10,5)

y<-c(23,16,10,4,22,15,12,7,21,16,11,5,22,14,10,6)
```

The elements of x and y are paired because the 2 samples were taken on the same river, upstream (*y*) or downstream (*x*) of a sewage outfall. If we ignore the fact that the samples are paired, it appears that the sewage outfall has no impact on biodiversity score ($p = 0.6856$):

```
t.test(x,y)

     Standard Two-Sample t-Test

data:  x and y
t = -0.4088, df = 30, p-value = 0.6856
alternative hypothesis: true difference in means is not
equal to 0
95 percent confidence interval:
 -5.246747  3.496747
sample estimates:
 mean of x mean of y
      12.5    13.375
```

However, if we allow that the samples are paired (simply by specifying the option **paired=T**), the picture is completely different.

```
t.test(x,y,paired=T)

     Paired t-Test

data:  x and y
t = -3.0502, df = 15, p-value = 0.0081
```

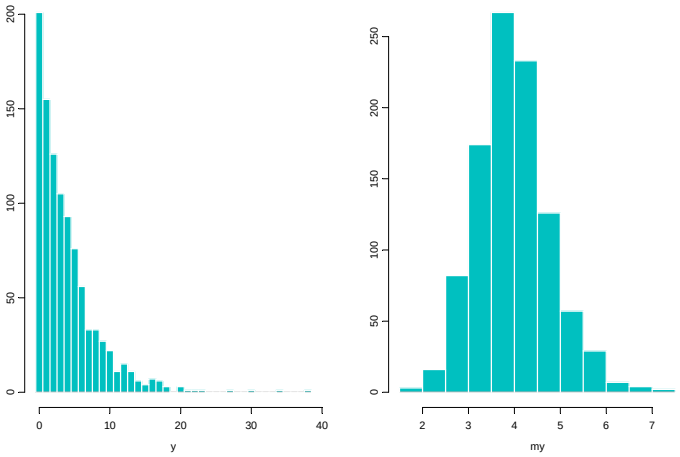
```
alternative hypothesis: true mean of differences is not
equal to 0
95 percent confidence interval:
 -1.4864388 -0.2635612
sample estimates:
 mean of x - y
       -0.875
```

The difference between the means is highly significant (p = 0.0081). The moral is clear. If you have information on blocking (the fact that the two samples came from the same river in this case), then use it in the analysis. It can never do any harm, and sometimes (as here) it can do a huge amount of good.

Central Limit Theorem

The central limit theorem states that for any distribution with a *finite variance*, the mean of a *random sample* from that distribution tends to be normally distributed.

The central limit theorem works remarkably well, even for really badly behaved data. Take this negative binomial distribution that has a mean of 3.88, a variance mean ratio of 4.87, and k = 1.08 (on the left): you see that the means of repeated samples of size n = 30 taken from this highly skew distribution are close to normal in their



distributions. The panels were produced like this. The left hand histogram is a frequency distribution of 1000 negative binomial random numbers (with parameters size = 1, probability = 0.2). The frequency distribution is viewed using **table**. There were 201 zero's in this particular realisation, and 1 value of 38. Note the use of the sequence to produce the required **break** points in the histogram.

```
par(mfrow=c(1,2))
y<-rnbinom(1000,1,.2)
```

This says generate 1000 random numbers from a negative binomial distribution whose 2 parameters are 1.0 and 0.2 respectively.

```
mean(y)

[1] 3.879
var(y)

[1] 18.90326

table(y)

  0   1   2   3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 20 21 22 23 27 30 34 38
201 155 126 105 93 76 56 33 33 27 22 11 15 11  6  4  7  6  3  3  1  1  1  1  1  1  1

hist(y,breaks=-0.5:38.5)
```

The right hand panel shows the distributions of means of samples of n = 30 from this same negative binomial distribution. One thousand different samples were taken and their average recorded in the vector called *my* from which the histogram is produced. When you type the for loop, hit “hard return” after the opening curly bracket and at the end of each line inside the multi-line function (you will see the line continuation prompt “+” at the beginning of each line until the closing curly bracket is entered).

```
my <- numeric(1000)
for (i in 1:1000) {
                y <- rnbinom(30, 1, 0.2)
                my[i] <- mean(y) }
hist(my)
```

The 1000 calculations are carried out very swiftly, and you see a normal distribution of the mean values of y. In fact, the central limit theorem works reasonably well for sample sizes much smaller than 30 as we shall see in a later practical.

```
par(mfrow=c(1,1))
```