3 Nonparametric Methods

3.1 Introduction

We continue with the idea of hypothesis testing, but consider alternative nonparametric approaches which rely on considering *permutations* of the data. We first consider permutation tests, in which we generate all possible equally-likely permutations of the data, assuming that the Null hypothesis is true, and then examine whether the single permutation that we observed is extreme when compared with the other possible permutations that might have arisen. By permuting the observed data, we emulate the distribution of the test statistic under the Null hypothesis. These methods, together with an appropriate choice of test statistic, lead to some of the classical nonparametric tests: the sign test, Wilcoxon's signed ranks test, and the Mann-Whitney test. The advent of modern powerful computers allows us much greater flexibility in choice of test statistics, and we show how randomization tests, in which the computer is used to generate a random subset of all possible permutations, are used in some familiar settings.

We will need some definitions for measurement scales of data:

Nominal: This is for when our data are simply labels. For example, rocks can be generally categorized as igneous, sedimentary and metamorphic.

Ordinal: If data can be rank-ordered, then their scale is said to be ordinal. Ordinal measurements describe order, but not relative size or degree of difference between the items measured. For example, in a questionnaire, respondents might be asked to rate a lecturer as 'bad', 'average' or 'good'.

Interval: Quantitative attributes are all measurable on interval scales, as any difference between the levels of an attribute can be multiplied by any real number to exceed or equal another difference. An example of interval scale measurement is temperature in degrees centigrade. The 'zero point' on an interval scale is arbitrary, and negative values can be used.

Ratio: Most measurement in the physical sciences and engineering is done on ratio scales. Mass, length, time, plane angle, energy and electric charge are examples of physical measures that are ratio scales. The distinguishing feature of a ratio scale is that the zero value is not arbitrary. For example, the Kelvin temperature scale has a non-arbitrary zero point of absolute zero.

In practice, observations are usually treated as either categorical (nominal, ordinal) or continuous (interval, ratio). Permutation and randomization tests are nonparametric in that we do NOT assume an underlying distribution for the process that generated the observed data. Some of the reasons why one may want to consider nonparametric tests are:

- Continuous measurements clearly do not follow the Normal distribution
- Outliers are present; even if the vast majority of the observations are in accordance with the Normal distribution, this may influence statistical inferences.
- The number of observations is too small to assess Normality.
- Observations are not continuous measurements

3.2 The Two-Sample Permutation Test

Using the sample means to construct the test statistic

The concept of a randomization test is best illustrated by example. Suppose we have two samples:

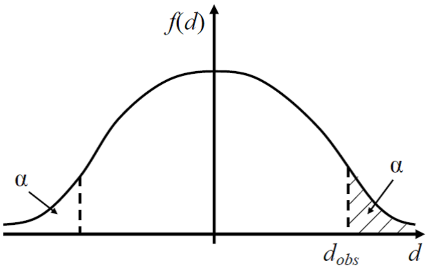
x	8	6	3	9		
y	7	10	10	12	18	15

and we wish to test:

$$H_0\text{: the two distributions from which the samples were drawn are the same}$$

vs.

$$H_1\text{: the two distributions are different.}$$



Typically, we are interested in whether the two distributions have different underlying means.

Let μ_X and μ_Y denote the corresponding mean for X and Y, respectively. Then, under H_0, $\mu_X = \mu_Y$; and under H_1, $\mu_X \neq \mu_Y$. Estimate μ_X and μ_Y by $\bar{x}$ and $\bar{y}$ and consider the test statistic: $d = \bar{y} - \bar{x}$. Under H_0, $d \approx 0$; whereas under H_1, d is not ≈ 0.

The idea underlying the permutation test is as follows. If we "permute" the data randomly between the two groups (keeping each group size fixed), then under H_0, the observed dataset should be a typical member of these "permuted" datasets. In this example we can consider all possible datasets (this is called a permutation test). So we need to identify the possible datasets which result in a value of d that is at least as extreme as the test statistic for the observed dataset, d_{obs}, to calculate the p-value.

However, we have not defined what we mean by "at least as extreme as" for a two-tailed test. Suppose that our observed test statistic d_{obs} is the upper $100(1 - \alpha)\%$ quantile. This means that the probability of observing a value at least as large as d_{obs} is α, or $100\alpha\%$ under the null hypothesis. Similarly, a test statistic would be just as extreme if it appeared in the lower $100\alpha\%$ of the distribution.

Hence, if H_0 is true, the probability of observing a value at least as extreme as the observed statistic is 2α. Thus, we only need to calculate the one-tailed probability of getting a value of d at least as extreme as d_{obs}. For a two-tailed test, we double this value.

Returning to our example, we begin by calculating the test statistic d for the observed data, giving 5.5. We now need to consider all permutations of the data which give a value for d of 5.5 or more. This is equivalent to finding all samples of size 4 that give a sample total at most as big as the total of x's in the real sample. That sample comprised the values 3, 6, 8, 9. Only two samples give a smaller total: 3, 6, 7, 8 and 3, 6, 7, 9. There are also three that give the same total: the original sample, together with two samples comprising the values 3, 6, 7, 10. (There are two 10's in the combined sample, explaining why there are two of these.) Thus, there are five possible combinations with a test statistic at least as large as the observed test statistic.

x				y						$\bar{x}$	$\bar{y}$	$d = \bar{y} - \bar{x}$
3	6	8	9	7	10	10	12	15	18	6.5	12	5.5
3	6	7	8	9	10	10	12	15	18	6	$12\frac{1}{3}$	$6\frac{1}{3}$
3	6	7	9	8	10	10	12	15	18	6.25	$12\frac{1}{6}$	$5\frac{11}{12}$
3	6	7	10	8	9	10	12	15	18	6.5	12	5.5
3	6	7	10	8	9	10	12	15	18	6.5	12	5.5

In total, there are $\binom{10}{4} = 210$ possible combinations, all of which are equally likely under the null hypothesis. Hence the one-tailed probability of obtaining a test statistic at least as large as the observed one is $\frac{5}{210}$.

For a two-tailed test, the corresponding p-value is $\frac{10}{210} = 0.0476$. Thus we have moderate evidence to reject H_0 — we reject H_0 at the 5% significance level.

For a one-tailed alternative hypothesis:

$$H_0\colon \mu_X = \mu_Y$$
vs.
$$H_1\colon \mu_X < \mu_Y$$

then we need only consider the cases where $d(= \bar{y} - \bar{x}) \geq 5.5$, and so the p-value is $\frac{5}{210} = 0.024$.

Using the ranks of the observations to construct the test statistic

Assume continuous observations. Suppose we have two independent random samples $x_1, x_2, \cdots, x_{n_1}$ and $y_1, y_2, \cdots, y_{n_2}$. We wish to assess whether the two samples come from the same distribution. For example, we may wish to compare two treatments, to assess whether one has a greater effect than the other.

The test is focused on a possible difference in location between the two distributions. In addition to assuming that the two samples are independent, we assume that the distribution functions for the two samples are both continuous; if they are not, the test can still be used, but the result tends to be conservative.

If we assume that the distributions differ in location only, the Null hypothesis is: $H_0 : M_X = M_Y$; that is, the medians of the two samples are the same. The alternative might be one-tailed ($H_1 : M_X > M_Y$ or $H_1 : M_X < M_Y$) or two-tailed ($H_1 : M_X \neq M_Y$). We could also consider that the hypotheses are in terms of the means, *e.g.* $H_0 : \mu_X = \mu_Y$. More generally, we can consider the null hypothesis to be that the two distributions are identical, although the test has no power for testing for differences between distributions if their location is the same.

To carry out the test, we pool the two samples, and list the observations in increasing order. Replace each observation by an x or a y, according to which sample it belonged. For each x_i, evaluate $v_i =$ number of y observations which are less than x_i.

Our test statistic is $U_X = \sum_{i=1}^{n_1} v_i$.

Thus, for the pooled data sequence, the statistic U_X gives the total number of times a y observation precedes an x observation. This test is also known as the Mann-Whitney U Test, or the Wilcoxon Rank Sum Test.

Example

Consider the data used to illustrate the two-sample permutation test earlier:

x	8	6	3	9		
y	7	10	10	12	18	15

We wish to test:

H_0: the two distributions from which the samples were drawn are the same
vs.
H_1: the two distributions are different.

If we place the observations in increasing order, we have:

```
3 6 7 8 9 10 10 12 15 18
```

Replacing observations by an x if they are from the first sample, or a y otherwise, gives:

```
x x y x x y y y y y
```

Note that the tie in these data causes no difficulty, because both 10's occur in the same sample. Hence there is no ambiguity in the value of the test statistic.

We have four x's. No y's are less than two of these, and a single y is less than the other two. Hence we have $U_X = 0 + 0 + 1 + 1 = 2$. We can list every possible ordering of 4 x's and 6 y's, and determine the proportion of these that give a test statistic equal to 2 or less. The following permutations meet this condition:

```
                        Test statistic
x x x x y y y y y y           0
x x x y x y y y y y           1
x x y x x y y y y y           2
x x x y y x y y y y           2
```

The total number of permutations is $\begin{pmatrix} 10 \\ 4 \end{pmatrix} = 210$, giving us a $p-$value of $4/210$ for a one-tailed test of whether the location of the x distribution is less than that of the y distribution. We have a two-tailed alternative hypothesis, so our $p-$value is $8/210 = 0.038$ or 3.8%. This compares with 4.76% that we obtained from our earlier two-sample permutation test.

In practice, we do not need to enumerate the permutations that are at least as extreme as the observed one, as we can look up the Mann-Whitney U statistic in published tables, or use R (below).

Comments

The value that U_X takes can lie anywhere between 0 (when the smallest n_1 observations all come from sample 1) and $n_1 n_2$ (when the smallest n_2 observations all come from sample 2). The distribution is symmetric about $n_1 n_2/2$ if the null hypothesis holds. Hence $E(U_X) = n_1 n_2/2$ if H_0 is true. It can be shown that the variance under H_0 is $n_1 n_2(n_1+n_2+1)/12$. For large n_1 and n_2, we can use these results together with the CLT to obtain a test based on a normal approximation. A continuity correction improves this approximation.

If the two samples are normally distributed with the same variance, then this permutation test is slightly less powerful than the two-sample $t-$test. In general, it can also be expected to be less powerful than the two-sample permutation test based on the sample means. It loses power by throwing away the actual values of the observations, but the power loss is modest. The advantage is that outliers do not affect statistical inferences disproportionately.

Using R

The test can be carried out using the 'wilcox.test' function. The package uses the statistic U_X but denotes it by W. Hypotheses are expressed for the case that the two distributions differ only in location. The default, if there are no ties and the sample sizes are less than 50, is to calculate exact $p-$values. Otherwise, a normal approximation is used.

For our example, we have

```
> x<-c(8,6,3,9)
> y<-c(7,10,10,12,18,15)
> wilcox.test(x,y,paired=FALSE,alternative="two.sided")  # or use "greater" or "less"

        Wilcoxon rank sum test with continuity correction

data:  x and y
W = 2, p-value = 0.0422
alternative hypothesis: true location shift is not equal to 0

Warning message:
In wilcox.test.default(x, y, paired = FALSE, alternative = "two.sided") :
  cannot compute exact p-value with ties
```

Why does this give a different result than we obtained previously? The clue is in the warning. R has detected a tie, and has therefore switched to a normal approximation. But this doesn't make sense, as the tie is within one of the samples, and therefore does not affect the test! To verify, let's replace one of the 10's by 11, which should not alter our result:

```
> x<-c(8,6,3,9)
> y<-c(7,10,11,12,18,15)
> wilcox.test(x,y,paired=FALSE,alternative="two.sided")

        Wilcoxon rank sum test

data:  x and y
W = 2, p-value = 0.03810
alternative hypothesis: true location shift is not equal to 0
```

We now get the same result as previously. The programmers of R are clearly not infallible!

3.3 Permutation Tests for Matched Pairs

Suppose we have n pairs of data, $(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)$. These might be for example before and after readings on a set of n individuals, or measurements on n pairs of individuals. In the latter case, pairs are selected so that individuals within a pair are as alike as possible, to reduce experimental error. For example, two hand creams might be tested by applying one to the left hand and the other to the right hand of an individual, or in a treatment for an illness, individuals might be selected as a pair on the basis of similar age, severity of illness, social background, etc.

Typically, analysis of matched pairs is conducted by first calculating the differences, $d_i = y_i - x_i$, $i = 1, \cdots, n$. If these differences can reasonably be assumed to be normally distributed with constant but unknown variance, then we can used a paired $t-$test of $H_0: \ \mu_X = \mu_Y$ against a one- or two-sided alternative (Section 5 of the notes). We now consider options for when the data are not normally distributed.

We cannot now generate permutations of the data by reallocating the pooled set of observations to the two samples, because this fails to preserve the pairs. It assumes that we have two independent samples, but we know that observations within a pair are not independent. Instead, we again work with the differences d_i, which can be assumed to be independent.

Example

We have 400m race times for 8 athletes competing at both sea level and altitude:

Runner:	1	2	3	4	5	6	7	8
Time at sea level:	48.3	47.6	49.2	50.3	48.8	51.1	49.0	48.1
Time at altitude:	50.4	47.3	50.8	52.3	47.7	54.5	48.9	49.9
Difference:	-2.1	0.3	-1.6	-2.0	1.1	-3.4	0.1	-1.8

We now generate all permutations by considering $\pm d_i$ for $i = 1, \cdots, n$:

```
  -0.1 -0.3 -1.1 -1.6 -1.8 -2.0 -2.1 -3.4
   0.1 -0.3 -1.1 -1.6 -1.8 -2.0 -2.1 -3.4
  -0.1  0.3 -1.1 -1.6 -1.8 -2.0 -2.1 -3.4
   0.1  0.3 -1.1 -1.6 -1.8 -2.0 -2.1 -3.4
...
   0.1  0.3  1.1 -1.6 -1.8 -2.0 -2.1 -3.4
...
   0.1  0.3  1.1  1.6  1.8  2.0  2.1  3.4
```

Different choices for the test statistic lead to different tests. We consider standard options below.

Using the sample mean as test statistic

Assume that the measurements are continuous observations. Consider

$$H_0\text{: } \mu_x = \mu_y$$
vs.
$$H_1\text{: } \mu_x \neq \mu_y.$$

We have $d_i = y_i - x_i$ and set $z_i = -d_i$ for $i = 1, \ldots, n$. If H_0 is true and we can assume symmetry, then we are as likely to observe z_i as d_i. Thus we generate permutations by taking either d_1 or z_1 with either d_2 or z_2, and so on, giving 2^n permutations in all. We take the test statistic to be the mean of the differences, and generate all of the 2^n possible permutations, evaluating the mean difference for every permutation. The p-value is the number of permutations with a test statistic at least as extreme as that obtained from the original data.

Example

Athletes' 400m times:

Runner:	1	2	3	4	5	6	7	8
Time at sea level:	48.3	47.6	49.2	50.3	48.8	51.1	49.0	48.1
Time at altitude:	50.4	47.3	50.8	52.3	47.7	54.5	48.9	49.9
Difference:	-2.1	0.3	-1.6	-2.0	1.1	-3.4	0.1	-1.8

Suppose we wish to test $H_0: \ \mu_d = 0$ against $H_1 : \mu_d < 0$, and we use the mean difference as our test statistic. We can list all permutations, and evaluate our test statistic for each:

```
  -0.1 -0.3 -1.1 -1.6 -1.8 -2.0 -2.1 -3.4    :    -1.55
   0.1 -0.3 -1.1 -1.6 -1.8 -2.0 -2.1 -3.4    :    -1.525
  -0.1  0.3 -1.1 -1.6 -1.8 -2.0 -2.1 -3.4    :    -1.475
   0.1  0.3 -1.1 -1.6 -1.8 -2.0 -2.1 -3.4    :    -1.45
...
   0.1  0.3  1.1 -1.6 -1.8 -2.0 -2.1 -3.4    :    -1.175
...
   0.1  0.3  1.1  1.6  1.8  2.0  2.1  3.4    :     1.55
```

We then count up the number of permutations with a mean less than or equal to the sample mean, -1.175. This number divided by $2^8 = 256$ gives our p-value. If the alternative hypothesis was two-sided, we would double this p-value.

Using the ranks of the observations to construct the test statistic (Wilcoxon's signed ranked test)

Assume that the measurements are continuous observations. In some circumstances, continuous observations may be very noisy, and we may wish to adopt a test statistic that is largely unaffected by outliers. A common strategy for nonparametric tests is to replace the observations by their ranks. Tests based on ranks exploit the relative size of observations, but do not use their actual values. For this test, we rank the absolute values of the observations from smallest to largest, ignoring their sign, then assign the sign of each observation to the corresponding rank. A permutation test is then conducted, using as test statistic the sum of the positive ranks.

We assume that the random variables D_i are independent and have a common median M. The measurement scale must be continuous, and the differences D_i must be symmetrically distributed about M. (In fact, the test can be applied to ordinal data, but the concept of symmetry is not then meaningful.) Given that symmetry is assumed, then the median is equal to the mean. Thus, although the null hypothesis is still generally stated in terms of the median, the test could be used as a test that the mean of the differences is zero.

We discard any pairs for which the difference is zero. For our example, we have:

Runner:	1	2	3	4	5	6	7	8
Time at sea level:	48.3	47.6	49.2	50.3	48.8	51.1	49.0	48.1
Time at altitude:	50.4	47.3	50.8	52.3	47.7	54.5	48.9	49.9
Difference:	-2.1	0.3	-1.6	-2.0	1.1	-3.4	0.1	-1.8
Absolute difference:	2.1	0.3	1.6	2.0	1.1	3.4	0.1	1.8
Rank:	7	2	4	6	3	8	1	5
Signed rank:	-7	2	-4	-6	3	-8	1	-5

Hence our test statistic is $T^+ = 2 + 3 + 1 = 6$.

Note that in general, T^+ can take any value between 0 and $r(r + 1)/2$ (i.e. the sum of the first r positive integers). Here, with $r = 8$, T^+ can take any value between 0 and 36.

We have $H_0 : M = 0$. If we had $H_1 : M > 0$, then we would have no evidence to reject H_0, as the sample median is actually negative. Now consider $H_1 : M < 0$. Is our sample median significantly less than zero? Again, we could list all possible permutations, taking plus or minus each rank: $\pm 1, \pm 2, \cdots, \pm 8$. We could then find the proportion of these permutations that give a value for the test statistic of at most 6. Alternatively, we can simply use published tables. Or we can use R:

```
> x <- c(48.3,47.6,49.2,50.3,48.8,51.1,49.0,48.1)
> y <- c(50.4,47.3,50.8,52.3,47.7,54.5,48.9,49.9)
> wilcox.test(x,y,paired=TRUE,alternative="less")

        Wilcoxon signed rank test

data:  x and y
```

```
V = 6, p-value = 0.05469
alternative hypothesis: true location shift is less than 0

>
> # Other options:
>
> wilcox.test(x,y,paired=TRUE,alternative="two.sided")

        Wilcoxon signed rank test

data:  x and y
V = 6, p-value = 0.1094
alternative hypothesis: true location shift is not equal to 0

> wilcox.test(x,y,paired=TRUE,alternative="greater")

        Wilcoxon signed rank test

data:  x and y
V = 6, p-value = 0.961
alternative hypothesis: true location shift is greater than 0
```

Hence for our one-sided test, the $p-$value is just above 0.05. We have weak evidence that the median is not zero; there is an indication that run times may be faster at altitude. If we were to conduct a two-sided test, our $p-$value doubles, as expected.

We can find the expectation and variance of T^+ under the null hypothesis by writing $T^+ = \sum_{i=1}^r \delta_i.i$ where $Pr(\delta_i = 1) = Pr(\delta_i = 0) = 0.5$. Hence if the null hypothesis is true, $E[\delta_i] = 0 \times 0.5 + 1 \times 0.5 = 0.5$, $E[\delta_i^2] = 0^2 \times 0.5 + 1^2 \times 0.5 = 0.5$, and $\text{var}[\delta_i] = E[\delta_i^2] - E[\delta_i]^2 = 0.5 - 0.5^2 = 0.25$. Hence under H_0, $E[T^+] = \sum_{i=1}^r E[\delta_i].i = 0.5 \sum_{i=1}^r i = r(r+1)/4$ and $\text{var}[T^+] = \sum_{i=1}^r \text{var}[\delta_i].i^2 = 0.25 \sum_{i=1}^r i^2 = r(r+1)(2r+1)/24$. For large sample sizes, this allows us to use a normal approximation:

$$\frac{T^+ - r(r+1)/4}{\sqrt{r(r+1)(2r+1)/24}} \sim N(0,1) \tag{8}$$

approximately under H_0.

Here, the approximation is improved using a **continuity correction**: reduce the magnitude of the numerator by 0.5.

Critical values in published tables assume that there are no **ties**. That is, we assume that no two differences are identical in magnitude. The effect of ties is slight when the normal approximation is used. In R, if there are no ties, the exact $p-$value is calculated for $n < 50$, but the default is to use the normal approximation whatever the sample size if ties are present.

This test is based only on the ranks of the differences. It uses less information compared to using the actual observed values to calculate the sample mean. Therefore, it is less powerful when interval or ratio data are available without any outliers.

Using the sign of the differences to construct the test statistic (Sign test)

Now, the test statistic is simply the number of positive differences. Note that to apply this test, we merely need to state whether, within each pair, x_i is smaller or larger than y_i, so that the measurement scale must be ordinal, scale or ratio.

Typically, this test is used to test $H_0: M = 0$ against a one- or two-sided alternative, where M is the median of the differences d_i. If we denote the random variable corresponding to d_i by D_i, $i = 1, \cdots, n$, then we assume that the D_i are independent with a common median M. For continuous data, it follows from the definition of a median that $Pr[D_i > M] = Pr[D_i < M] = 0.5$. For discrete data, $Pr[D_i = M]$ may exceed zero; in this case, we must assume that $Pr[D_i > M] = Pr[D_i < M]$.

If the X and Y distributions are the same, we expect the differences d_i to be centred around zero. We discard any differences that equal zero; suppose that this leaves r pairs, which we relabel as $d_1, \cdots, d_r$. Our test statistic is $S^+ =$ the number of differences that are positive.

For our example, we have $n = r = 8$ and $S^+ = 3$. To assess whether this is an extreme result, we simply need to evaluate the number of positive differences for each of the $2^8 = 256$ permutations, $\pm d_1, \pm d_2, \cdots, \pm d_8$. Fortunately, there is a quicker method! We have 8 independent trials, and under H_0, the probability of a 'success' (i.e. a positive difference) is 0.5. Hence if H_0 is true, S^+ is simply an observation from a binomial distribution with 8 trials (in general, r trials) and probability of success $p = 0.5$.

If we have $H_1 : M < 0$, then the $p-$value of our test is given by $Pr[S^+ \leq 3]$, which we can evaluate in R:

```
> pbinom(3,8,0.5)
[1] 0.3632813
```

Alternatively, we can use published tables. For large r, we can use a normal approximation to the binomial distribution.

This test is easy to perform but, as it is based only on the signs of the differences, it wastes information and is less powerful when interval or ratio data are available, compared to using the sample mean or the ranks of the observations.

3.4 One sample permutation tests

A **one-sample permutation test** of $H_0 : \mu_y = \mu_0$ is conducted in exactly the same way, except that the differences d_i are replaced by $y_i - \mu_0$, where y_i are our observations, $i = 1, ..., n$.

3.5 Computer-intensive Randomization

Tests based on ranks are convenient because the number of cases for small sample sizes is sufficiently limited that tables of critical values can be published. (Different samples of size n all have exactly the same set of ranks, 1 to n, provided there are no ties.) Also, the tests can be conducted in R with minimal programming. By contrast, permutation tests based on the original observations are not included in the standard R package, it is far from trivial to program the tests to generate every possible permutation, and even for quite small sample sizes, the amount of computation required quickly becomes excessive.

Suppose that we observe two samples $x_1, \ldots, x_m$ and $y_1, \ldots, y_n$. Then, if $m = 10$ and $n = 10$, there are only 20 observations in total. However there are 184756 possible combinations with 10 observations in each sample. It can be difficult to identify all the combinations which give a value at least as extreme as the test statistic from the observed data. Suppose we double our sample size for each group. We would then have a total of 40 observations and 1.38×10^{11} possible combinations for dividing the observations into two groups of 20. If we calculated one combination per second, it would take 4371 years to enumerate them. Even a computer working a million combinations per second would still take 38 hours to enumerate all possible combinations. Clearly it is not always feasible to consider all possible permutations.

The solution to this problem is to generate a more reasonable number of combinations independently and randomly. Provided that all possible combinations have the same probability of being generated, this random sample of combinations can be used to generate the approximate distribution of the test statistic under H_0. Thus, to perform a test, generate r independently randomized replicates and calculate the test statistic for each. Add the original test statistic to the list (after all, under the null hypothesis it is a perfectly valid value obtained from a randomization). Now calculate the proportion of the $r + 1$ values that are at least as extreme as the observed test statistic; this gives the estimated $p-$value of the test.

We need computer code to conduct this test. The code is best incorporated into an R function. A simple way to write the code is to use Word, Notepad or some other editor. This allows you to write and edit your code, and then you can paste it into R, or use the "source" command for R to read the file. If there is an error, you can then return to the code in your editor, correct the mistake, and paste/read the code back into R. You can also save the code in a file in your workspace, for future use. Consider again the two-sample problem above, where we wish to test whether the two observed samples are from the same distribution, but it is not feasible to enumerate all permutations at least as extreme as the observed one. For example, consider the following data on the production of nitrogen-bound serum albumen in diabetic and non-diabetic mice:

Diabetic: 391,46,469,86,174,133,13,499,168,62,127,276,176,146,108,276,50,73
Non-diabetic: 156,282,197,297,116,127,119,29,253,122,249,110,143,64,26,86,122,455,655,14

There are 3.36 $\times 10^{10}$ possible combinations, and it is clear that many will give a test statistic (difference in sample means) at least as extreme as the observed test statistic.

We wish to test the hypothesis:

$$H_0\colon \mu_X = \mu_Y$$
$$\text{vs.}$$
$$H_1\colon \mu_X \neq \mu_Y,$$

where μ_X and μ_Y are the underlying means for diabetic and non-diabetic mice, respectively. We use the test statistic $d = \bar{x} - \bar{y}$. We then use R to generate independent random permutations of the data and use these to obtain an approximate p-value.

R code

The following code assumes that the data are in a column labelled `level`, with an additional column `index`, which indexes whether the observation is for a diabetic ('1') or non-diabetic ('2') mouse. The key purpose of the code is to randomize the order of the observations, while leaving the index values unaltered, as the effect of this is to split at random the pooled data into two samples of the required sizes. Each random split comprises one randomization.

```
serum <- read.table("mouse.txt",header=T)      # Read data from file mouse.txt
attach(serum)                                  # This tells R to look in data frame
                                               # 'serum' for variables; it saves
                                               # having to type e.g. 'serum$level'
nrand <- 9999                                  # set nrand equal to 9999
teststat <- mean(level[index=="1"])-           # Calculate the test statistic
   mean(level[index=="2"])
n <- length(level)                             # Calc total number of observations

rand.function <- function (nrand) {            # Define the function 'rand.function'
r <- vector (length=nrand+1)                   # Define a vector r of length nrand+1
for (i in 1:nrand){                            # Implement the code in {} for
                                               # i=1,2,3,...,nrand

randlevel <- sample (level, n, replace=F)      # Randomly reorder the n obsns in level
randmean <- mean(randlevel[index=="1"])-       # Calculate the test statistic for
   mean(randlevel[index=="2"])                 # each randomization
r[i+1] <- randmean}                            # Put test stat for randomization i
                                               # into element i+1 of vector r

r                                              # An R function returns whatever is
                                               # output in the last line; in this
}                                              # case, the vector of test stats

results <- rand.function(nrand)                # Call rand.function with nrand random-
                                               # izations and put output in 'results'
results[1] <- teststat                         # Add 'teststat' to the top of the list
p <- length(results[results>=teststat])/       # Or we can calculate the proportion at
   length(results)                             # least as big as 'teststat' ...
if (p>0.5) {p <- length(results                # and look in the other tail if
   [results<=teststat])/length(results)}       # p>0.5 ...
p <- 2*p                                       # and double it for a 2-tailed test ...
if (p>1) p <- 1                                # can find p>1 in which case should be 1
p                                              # and output the p-value
```

The last few lines calculate the $p-$value. It would be perfectly acceptable simply to order the results (`sort(results)`), and pick out how many values are at least as extreme as the observed test statistic.

Note also that the function has looked at only one tail of the distribution under the null hypothesis, so that the $p-$value must be doubled if the alternative hypothesis is two-tailed as here.

When I ran this macro on the mouse data, the $p-$value against a two-tailed alternative was 0.98. The data provide no evidence against the null hypothesis. In fact, by chance the two samples have almost the same mean, and almost all randomizations show a larger difference between means than do our real data.

When I used `nrand` = 99 and ran the code a number of times, I obtained p-values ranging from 0.88 to 1. Increasing `nrand` to 999, I obtained p-values in the range (0.92, 0.98); for `nrand` = 9999, approximate p-values were in the range (0.97, 0.99); for `nrand` = 99999, I consistently obtained a p-value of 0.98. By increasing the number of randomizations of the data we obtain more consistent estimates of the p-value — this is because we are evaluating the test statistic for more of the possible permutations.

3.6 Multiple-sample Randomization Test

Suppose we now have three (or more) independent samples. Then, we wish to test:

$$\begin{gathered} H_0\text{: all distributions are the same} \\ \text{vs.} \\ H_1\text{: at least one distribution is different.} \end{gathered}$$

We assume that we are interested in detecting differences in the means of the populations. The underlying idea remains the same. Under H_0, we can reshuffle the observations between groups and the observed data will appear similar to those simulated if H_0 is true. In order to do this we need to define a single test statistic t.

We let $\bar{x}$ be the overall mean and $\bar{x}_i$ the mean of group i. Then we could use

$$t = \sum_i (\bar{x}_i - \bar{x})^2.$$

However, a better statistic would be

$$t = \sum_i n_i (\bar{x}_i - \bar{x})^2,$$

where n_i is the number of observations in group i (this is because $\text{var}(\bar{X}_i) \propto \frac{1}{n_i}$). Then, small values of t would support H_0; large values would support H_1.

Example

We observe data relating to the treatment of anorexia using three different treatments. We are interested in whether the different treatments produce different results in terms of weight gain of the individuals. The treatments are CBT (cognitive behavioural therapy), a "standard" treatment and family therapy. The data are the weight gains (in lb.) that resulted:

C.B.T.: 1.7, 0.7, -0.1, -0.7, -3.5, 14.9, 3.9, 17.1, -7.6, 1.6, 11.7, 6.1, 1.1, -4.0, 20.9, -9.1, 2.1, -1.4, 1.4, -0.3, -3.7, -0.8, 2.4, 12.6, 1.9, 3.9, 0.1, 15.4, -0.7
Standard: -0.5, -9.3, -5.4, 12.3, -2.0, -10.2, -12.2, 11.6, -7.1, 6.2, -0.2, -9.2, 8.3, 3.3, 11.3, 0.0, -1.0, 11.6, -4.6, -6.7, 2.8, 0.3, 2.0, 3.7, 5.9, 10.2
Family therapy: 11.4, 11.0, 5.5, 9.5, 13.6, -2.9, -0.1, 7.4, 21.5, -5.3, -3.8, 13.4, 13.1, 9.0, 3.9, 5.7, 10.7

A first step in an analysis is to assess whether there is any evidence that the efficacy of the treatments differ. In other words:

$$\begin{gathered} H_0\text{: all distributions are the same} \\ \text{vs.} \\ H_1\text{: at least one distribution is different.} \end{gathered}$$

We use the test statistic,

$$t = \sum_i n_i (\bar{x}_i - \bar{x})^2.$$

R code:

Assuming the data are in a single column labelled ‘gain’ of the file ‘anorexia.txt’, and a second column labelled ‘index’ takes values 1, 2 or 3, corresponding to the three treatments, the following R code could be used to implement a randomization test.

```
anorexia <- read.table("anorexia.txt",header=T)
attach(anorexia)
n1 <- length(gain[index=="1"])
n2 <- length(gain[index=="2"])
n3 <- length(gain[index=="3"])
n <- length(gain)

teststat <- n1*(mean(gain[index=="1"])-mean(gain))^2
teststat <- teststat + n2*(mean(gain[index=="2"])-mean(gain))^2
teststat <- teststat + n3*(mean(gain[index=="3"])-mean(gain))^2

multrand.function <- function(nrand) {
  r <- vector(length=nrand+1)
  for (i in 1:nrand){
    randgain <- sample (gain, n, replace=F)
    randstat <- n1*(mean(randgain[index=="1"])-mean(gain))^2
    randstat <- randstat + n2*(mean(randgain[index=="2"])-mean(gain))^2
    randstat <- randstat + n3*(mean(randgain[index=="3"])-mean(gain))^2
    r[i+1] <- randstat}
  r
}

nrand <- 9999
results <- multrand.function(nrand)
results[1] <- teststat
results <- sort(results)
p <- length(results[results>=teststat])/length(results)
p
```

Running the R code with nrand = 9999, I obtained a p-value of 0.022. Thus, there is evidence to reject H_0 in favour of H_1 — we reject H_0 at the 5% level.

3.7 A More General Way of Looking at Randomization

In a few lectures, it is not possible to show the full spectrum of randomization methods. For more information, see B.F.J. Manly, *Randomization, Bootstrap and Monte Carlo Methods in Biology*, Chapman & Hall, 1997. Here, an alternative way of looking at randomization is given which is quite helpful for approaching a wider range of testing problems.

In the randomization and permutation tests done so far, we have always talked about re-assigning data randomly to groups, but we could equally well have described the process as re-assigning the groups randomly to the data. This is particularly easy to see with reference to a randomization macro:

```
randlevel <- sample (level, n, replace=F)
```

This generates a single random permutation of the data in `level`; the variable `index` indicates the sample to which the randomized (randomly reordered) observations are assigned. We could instead randomize the index:

```
randindex <- sample (index, n, replace=F)
```

We now assign the observations to the two samples according to the index values in `randindex`. Both methods are equivalent: *it does not matter whether we permute the data or the labels identifying groups.* This observation motivates the following formulation.

Suppose that we have some measurements $y_1, \ldots, y_n$ of random variables $Y_1, \ldots, Y_n$, and that each Y_i is associated with some variables $x_{1i}, \ldots, x_{pi}$ which may influence the distribution of Y_i. The random variables $Y_1, \ldots, Y_n$ are often referred to as *response variables* and the x_{ji}'s as *explanatory variables.* For any given j (with $j = 1, \ldots, p$), if the distribution of each Y_i does not depend on $x_{j1}, \ldots, x_{jn}$ then the observed data should appear to be a typical member of the population of data sets obtainable by re-assigning $x_{j1}, \ldots, x_{jn}$ randomly to $y_1, \ldots, y_n$. As usual, 'appearance' will be judged by an appropriate test statistic. The randomization tests already discussed fall within this framework, as do the tutorial questions on randomization, but more complicated problems can also be addressed. For example, later in this module we shall consider fitting models of the form:

$$E(Y_i) = \alpha + \beta x_i + \gamma z_i$$

to data $y_1, \ldots, y_n$, where $x_1, \ldots, x_n$ and $z_1, \ldots, z_n$ are (values of) explanatory variables. The approach taken will be to find the values of the parameters α, β and γ which minimize the sum of squares of differences between the data $(y_1, \ldots, y_n)$ and the model, *i.e.* we minimize S, where

$$S = \sum_{i=1}^{n} (y_i - \alpha - \beta x_i - \gamma z_i)^2.$$

The minimized value ($\hat{S}$, say) of S can be used as a measure of how well the model fits the data. Now suppose that we want to test

$$\begin{gathered} H_0\colon\ \gamma = 0 \\ \text{vs.} \\ H_1\colon\ \gamma \neq 0. \end{gathered}$$

We can use $\hat{S}$ as a test statistic. Under the null hypothesis, the term in z_i does not offer any real improvement in model fit, so that the original $\hat{S}$ should be a typical element of the distribution of $\hat{S}$ values obtained by re-assigning $z_1, \ldots, z_n$ randomly to $y_1, \ldots, y_n$ and re-calculating $\hat{S}$ (by re-fitting the model). Performing such a hypothesis test by randomization gives a means for deciding whether or not there is sufficient evidence to retain any given explanatory variable in the model.

3.8 Matched-Pairs Randomization Test

The concept of reassigning observations to samples does not work for matched pairs. Suppose that we observe data $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$, and calculate differences $d_1, \ldots, d_n$, which are drawn from a distribution each with mean μ_d. Then our hypothesis test is:

$$\begin{gathered} H_0\colon\ \mu_d = 0 \\ \text{vs.} \\ H_1\colon\ \mu_d \neq 0. \end{gathered}$$

If we define $z_i = -d_i$, randomization is performed by choosing one observation from each pair (d_i, z_i) with equal probability of one half. We take the test statistic to be the mean of the differences, and simulate a large number of randomizations, calculating the mean of each generated permutation. The approximate p-value is the number of realizations with a test statistic at least as extreme as that obtained from the original data.

Example

Athletes' 400m times again:

```
Runner:                1    2    3    4    5    6    7    8
Time at sea level:   48.3 47.6 49.2 50.3 48.8 51.1 49.0 48.1
Time at altitude:    50.4 47.3 50.8 52.3 47.7 54.5 48.9 49.9
Difference:          -2.1  0.3 -1.6 -2.0  1.1 -3.4  0.1 -1.8
```

Suppose we wish to test $H_0 : \mu_d = 0$ against $H_1 : \mu_d < 0$, and we use the mean difference as our test statistic. We can use R to generate random permutations, by randomly permuting signs of the differences. For example, if the differences are in a variable called `diff`, we can generate a single randomization as follows:

```
n <- length(diff)
pm <- c(-1,1)
rdiff <- sample(pm,n,replace=T)*diff
```

If you are unsure what the above code does, enter the commands in R and try typing:

```
sample(pm,n,replace=T)
rdiff
```

The proportion of randomly-generated permutations that have a mean less than or equal to the mean of the original sample (-1.175) is the approximate *p*-value. For a two-sided alternative, we would double this *p*-value.

Exercise: Write a program to carry out a randomization test on the above differences. Compare your result with that obtained from the permutation test that uses signed ranks. Now replace the observations by their signed ranks, and use your code to verify the result from the permutation test. (You will need to generate a large number of randomizations, say 10,000 or 50,000, to ensure that Monte Carlo variation is very small.)