MT4113 Computing in Statistics

Background reading for lectures 6-7:
Computer-intensive statistical methods

Len Thomas
(based on notes by Steve Buckland)

1. Introduction

There is a preoccupation in applied statistics for testing irrelevant hypotheses. Too often, the perceived requirement for p-values takes precedence over estimation and common sense. Nonparametric tests are often used because the data are not normally distributed, as if they are therefore not amenable to more useful analysis. In this section of the course, we briefly explore the role of computer-intensive methods in data analysis. We show that simple methods exist that estimate parameters of interest, and quantify the precision of those estimates, even for messy real data sets that defy classical analysis.

Some of this material was already covered (in less depth) in MT2508 Stats Inference; if you're an undergrad then you should have taken that course as a pre-requisite to this one. (These ideas are also covered in MT3508 Applied Statistics, but that's not a pre-requisite.) If you're an MSc Applied Stats and Datamining or Data Analysis student, you'll have covered similar preparatory material in MT5756 Data Analysis. If you're an MSc Statistics student, then I hope you will have encountered basic nonparametric methods and computer-intensive methods before during your statistical training; if not and you are struggling with this material then please feel free to contact me for some extra support. Note for everyone: a copy of the MT2508 lecture notes are on the MT4113 Moodle site.

> Text that has a line down the left hand side (like this paragraph) can be considered as supplementary material; knowledge of such material is not essential to the course.

Two potentially useful texts are:

1. Davison, A.C. and Hinkley, D.V. 1997. *Bootstrap Methods and their Application.*
2. Manly, B.F.J. 1997. *Randomization, Bootstrap and Monte Carlo Methods in Biology.* 2nd edition.

Please note that the code in this document was not written by me, and does not conform to many of the principles of good programming practice that I've been teaching. I did not re-write it because I tend to ask for some of these methods to be programmed as assignments, and did not want to give the answers away! Hence, please do not take this code as an example of good practice. You may enjoy finding examples of bad practice as you look through it. ☺

2. One-sample methods

We start by introducing a motivating example that we'll use throughout the notes. As part of a study on pine martens in Kinlochewe, Scotland, radio tags were placed on 12 animals. Subsequent records of habitat usage allowed a habitat utilization index to be evaluated for each animal in several habitat types. The index was constructed so that a value of zero would be expected if an animal used a habitat type in proportion to its occurrence within the animal's territory. In the case of deciduous woodland, the following values were obtained:

0.13, -0.01, -0.01, 0.42, -0.02, 0.01, 0.09, 0.03, 0.04, 0.06, 0.12, 0.03

These values suggest that most animals use deciduous woodland roughly in proportion to its occurrence, but that one or two animals, especially animal four, may have a particular preference for the habitat. In this section, we use these data to show how to test the null hypothesis that the mean index value for the population represented by these 12 animals is zero, and how to quantify the precision of the estimate of this mean. Note that the sample size is very small, and many of the issues we discuss below (to do with checking for violation of assumptions, and appropriate transformations) would be eased with a larger sample size.

2.1 One-sample t-statistic

First consider the classical approach to this problem. There is an indication that the data are non-normal. This creates the classical analyst with a dilemma. Should she transform the data? How? It is difficult in this example to justify any particular transformation without reference to the data, yet the data themselves are too few to allow a satisfactory transformation to be estimated. If a transformation is identified, her problems are not over. What does she estimate the mean index value by? The untransformed sample mean? The back-transformed mean without bias correction? A bias-corrected back-transformed mean? Similarly, how does she back-transform variance and interval estimates? For illustrative purposes, we ignore these difficulties, and apply a one-sample Student's *t*-test, assuming that the untransformed index values are normally distributed with the same mean and variance.

Formalizing, let the mean index value for the population be represented by μ. We wish to test the null hypothesis $H_0 : \mu = 0$. The test is implemented by evaluating

$$t = \frac{\bar{x} - \mu}{\sqrt{s^2 / n}}$$

where n = the number of index values observed,

$\bar{x}$ = sample mean of the n index values = $\dfrac{\sum x_i}{n}$,

and s^2 = sample variance of the n index values = $\dfrac{\sum x_i^2 - \left(\sum x_i\right)^2 / n}{n-1}$.

We work through this by hand, for illustrative purposes – in practice you would likely let R do the calculations for you, using the function `t.test`. Thus we have

$n = 12$,

$$\bar{x} = \frac{0.13 - 0.01 - ... + 0.03}{12} = 0.07417,$$

$$s^2 = \frac{0.13^2 + (-0.01)^2 + \ldots + 0.03^2 - \dfrac{0.89^2}{12}}{11} = \frac{0.2235 - \dfrac{0.89^2}{12}}{11} = \frac{0.1575}{11} = 0.01432$$

Under $H_0 : \mu = 0$,

$$t = \frac{0.07417}{\sqrt{\dfrac{0.01432}{12}}} = 2.15$$

If the null hypothesis is true, this is a value from Student's t-distribution with degrees of freedom equal to n-1=11. Tables yield $t_{11}(0.10)$=1.80 (the value in brackets is the α-level for the test) and $t_{11}(0.05)$=2.20, so that, assuming the alternative hypothesis is two-tailed, the test statistic is significant at the 10% level but not the 5% level; there is only weak evidence that the mean utilization index for this population of pine martens differs significantly from zero. The data suggest that the population may preferentially utilize deciduous woodland, but the evidence is far from conclusive.

The same theory provides a 100(1-2α)% confidence interval for the true value μ, by setting

$$\frac{\bar{x} - \mu}{\sqrt{s^2 / n}} = \pm t_{n-1}(\alpha),$$

or

$$\mu = \bar{x} \pm t_{n-1}(\alpha)\sqrt{\frac{s^2}{n}} \; .$$

For our example, and α=0.025, this yields the following 95% confidence interval for μ:

$$0.07417 \pm 2.201\sqrt{\frac{0.01432}{12}}, \text{ or } (-0.002, 0.150).$$

(These calculations can be easily reproduced in R – the function `t.test` returns confidence intervals, or you can calculate the above formula directly, using the function `qt` to give $t_{n-1}(\alpha)$. Note that `qt` is looking for the upper quantile as the first argument, i.e., $1-\alpha/2$ for a two-sided confidence interval, so for the pine martin data you'd call `qt(0.975,11)`, which returns 2.200985.)

The assumptions of this method are essentially untestable for such a small sample. Further, it is not invariant to transformations of the data. If the arbitrary transformation $y_i = \log_e(x_i + 0.03)$ is applied, and a test carried out of $H_0 : \mu_y = \log_e(0.03) = -3.51$ (corresponding to utilization in proportion to occurrence), we obtain t=2.48. The back-transformed 95% confidence interval is

$$\exp\left(\bar{y} \pm t_{n-1}(0.05)\sqrt{\frac{s_y^2}{n}}\right) - 0.03 = \exp\left(-2.749 \pm 2.201\sqrt{\frac{1.115}{12}}\right) - 0.03,$$

or (0.003,0.095). Notice how much shorter this interval is.

2.2 Wilcoxon's signed rank test and confidence interval for the median

Is the user of the traditional nonparametric method on firmer ground? Wilcoxon’s signed rank test may be applied to single samples of data. The hypothesized median is subtracted from each observation. The resulting values are then ranked from smallest to largest, ignoring their signs, and then the signs are attached to the ranks. Although any monotone transformation of the observations leaves their ranks unaltered, this test *is* affected by transformation. This is because the signs of the observations are ignored when ranking them: e.g., in the above example, |-0.01| and |0.01| have the same rank, but $|\log_e(-0.01+0.03)-\log_e(0.03)|$ is ranked above $|\log_e(0.01+0.03)-\log_e(0.03)|$. The test is unaffected by changes in the sizes of the observations that leave the ranks unaltered. Thus in the above example, the largest observation, 0.42, could be replaced by any value from 0.14 to infinity, and the test statistic is unaffected. This is an advantage if inference based on the median is required, but the test is inappropriate if inference about the mean is required. Wilcoxon’s signed ranks test can be performed in R using the function `wilcox.test`.

Consider the above example, with $H_0 : M = 0$, where M is the median index value for the population represented by the sample of 12 animals. If the observations are ranked from smallest to largest, ignoring their sign, then the sign of the observation assigned to its rank, we get

Observation	0.13	-0.01	-0.01	0.42	-0.02	0.01	0.09	0.03	0.04	0.06	0.12	0.03
Rank of |obsn|	11	2	2	12	4	2	9	5.5	7	8	10	5.5
Signed rank	11	-2	-2	12	-4	2	9	5.5	7	8	10	5.5

If the null hypothesis is true, the sign attached to each rank is random, and the expectation of their sum is zero. Wilcoxon’s signed rank test is equivalent to applying a permutation test to the above signed ranks (see next section). It gives the exact *p*-value for the test of $H_0 : M = 0$ against the two-sided alternative $H_1 : M \neq 0$. In this case, we obtain $p = 0.012$.

A confidence interval for *M* may be obtained by testing $H_0 : M = m$ for a range of values for *m* (i.e., the standard “inverting the test statistic” or “test-inversion” approach to confidence interval generation). A 100(1-α)% confidence interval comprises all values of *m* that do not lead to rejection of H_0 in favour of a two-tailed alternative hypothesis at the 100α% level. Thus the confidence limits may be found using a search procedure. In our example, we obtain (0.010, 0.120).

2.3 Permutation test and permutation interval

Can we avoid the assumption that the observations are normally distributed, without reducing them to ranks? Consider first the null hypothesis that μ, the mean of the habitat utilization indices, is zero. If we are prepared to assume that the distribution of index values about their mean μ is symmetric (but not necessarily normal), then the sign of $x_i - \mu$ is random. Under $H_0 : \mu = 0$, all possible permutations of $\{\pm x_i\}$ are equally likely. If we enumerate them, we get:

0.13 0.01 0.01 0.42 0.02 0.01 0.09 0.03 0.04 0.06 0.12 0.03
-0.13 0.01 0.01 0.42 0.02 0.01 0.09 0.03 0.04 0.06 0.12 0.03
0.13 -0.01 0.01 0.42 0.02 0.01 0.09 0.03 0.04 0.06 0.12 0.03

0.13 0.01 -0.01 0.42 0.02 0.01 0.09 0.03 0.04 0.06 0.12 0.03

:

:

-0.13 -0.01 -0.01 -0.42 -0.02 -0.01 -0.09 -0.03 -0.04 -0.06 -0.12 -0.03

In all, there are $2^{12} = 4096$ different permutations. We need to determine how many of these are at least as extreme as the observed permutation of

0.13 -0.01 -0.01 0.42 -0.02 0.01 0.09 0.03 0.04 0.06 0.12 0.03

To do this, a suitable statistic must be selected. As the null hypothesis specifies a value for the population mean μ, an appropriate test statistic is $\bar{x}$. Under $H_0 : \mu = 0$, the farther from zero is $\bar{x}$, the more extreme is the permutation. The p-value is defined to be the probability that a value is at least as extreme as the observed statistic, given that the null hypothesis is true. Under H_0, all permutations are equally likely, so we simply need to evaluate the proportion of permutations that yield a sample mean $\geq \bar{x}$. If the alternative hypothesis is one-tailed, $H_1 : \mu > 0$, we only count positive values $\geq \bar{x}$ as at least as extreme, whereas if the alternative hypothesis is two-tailed, values $\leq -\bar{x}$ are also counted. (In fact, we usually just double the one-tailed p-value, as this is valid whether or not the test assumes symmetry about the mean.)

Although a very lengthy calculation if done by hand, we can enumerate each permutation, and evaluate the sample mean corresponding to each, on a computer. For the above data, we find that 48 permutations out of 4096 yield a mean ≥0.074 or ≤-0.074. If the alternative hypothesis is two-tailed, the p-value is thus 48/4096 = 0.012.

Now consider $H_0 : \mu = \mu_0$. We form the differences $0.13-\mu_0$, $-0.01-\mu_0$, ... , $0.03-\mu_0$, and proceed as before, enumerating all permutations $\{\pm(x_i - \mu_0)\}$. Suppose we have $H_1 : \mu > \mu_0$. We found above that for $\mu_0 = 0$ and a two-tailed alternative hypothesis, the p-value was 0.012. If we take one tail only, then the p-value is 0.006, and the null hypothesis is rejected at the 2.5% level. If we want a 95% **permutation interval**, we can slowly increase μ_0 from zero until the p-value (one-tailed test) just rises above 0.025. The corresponding value of μ_0 provides the lower 95% confidence limit. We can continue to increase μ_0 until a test of $H_0 : \mu = \mu_0$ against $H_1 : \mu < \mu_0$ yields a p-value below 0.025; the upper 95% confidence limit is given by the last value for which the p-value is at least 0.025. Because the permutation distribution is symmetric in this example, this 95% permutation interval comprises all values of μ_0 that are not rejected at the 5% level when $H_0 : \mu = \mu_0$ is tested against the alternative, $H_1 : \mu \neq \mu_0$. Standard search methods such as method of bisection (see later lectures on optimization) may be used to identify the limits. For our example, we obtain the interval (0.013, 0.150). Note that this interval is not symmetric about the estimate $\bar{x} = 0.074$.

If the same methods are applied to the transformed data $y_i = \log_e(x_i + 0.03)$, the test of $H_0 : \mu_y = \log_e(0.03) = -3.51$ yields a p-value of 0.030, and back-transforming the 95% permutation interval yields (0.003, 0.095). Thus this method is as sensitive to transformation as the method based on Student's t for the example data; the value of 0.42 is considerably larger than any other observation, and its effect is mitigated by the transformation, so that the upper confidence limit is appreciably reduced.

Sensitivity to a possible outlier can be reduced by other means. The permutation test has the advantage that it can be applied using an assortment of test statistics. Equivalent to the sample mean would be the sum of the observations, as the number of observations n is the same for every permutation. Other options include the sample median, trimmed means and the number of observations that are positive. Optimal choice of statistic depends on the null and alternative hypotheses that are of interest to the user.

If the mean is replaced by the median, the permutation test of $H_0 : M = 0$ yields a p-value of 0.023, and a 95% permutation interval for M is (0.010, 0.225). The high upper limit arises because the test is even more sensitive to the apparent asymmetry in the observed data than is the test based on the sample mean. Although the range of observations is from -0.02 to 0.42, one half of them are ≤0.03. For $M = 0.225$, 0.42 is further from the median than 0.03, and there are sufficient permutations of the values (-0.02 - M), (-0.01 - M), ... , (0.42 - M) that yield a more extreme sample median than the observed value of 0.035 to prevent the p-value dropping below 0.05. If the same test is applied to the transformed data, the back-transformed permutation interval for the median is (0.004, 0.076). Again, we have very different results depending on whether we transform the observations or not, with little information to judge what transformation, if any, is appropriate.

Another possible statistic is the trimmed mean, for which observations are trimmed from each end of the ordered list. If we delete just the smallest (-0.02) and largest (0.42) observations, and then apply permutation methods to the remaining ten, we obtain a 95% permutation interval for μ of (0.012, 0.086). Carrying out the same procedure on the transformed data and back-transforming yields (0.006, 0.078). Taking the median of the remaining ten observations as the test statistic, the corresponding intervals are (0.011, 0.080) and (0.005, 0.070) respectively. Alternatively, all 12 observations can be retained, and the trimming done within each permutation. Using the trimmed mean $\bar{x}_t = 0.041$ as the test statistic, we obtain the intervals (0.009, 0.074) and (0.032, 0.181), without and with transformation respectively. With the exception of the last approach, the differences between the methods are small after trimming.

If the test statistic is taken to be the number of values, (x_i - M), that are positive, the permutation test becomes the sign test. Thus, the test of $H_0 : M = 0$ yields a p-value of 0.146. A permutation interval may be found in the usual way. We obtain (-0.010, 0.120). These results are unaffected by any monotone transformation. The sign test avoids the assumption of symmetry; for continuous distributions at least, observations are equally likely to fall above the median as below by definition, whatever the true distribution. (If the sign test were to be used to test $H_0 : \mu = 0$, a symmetry assumption would be required, under which $\mu = M$.) The sign test statistic is the most robust of those suggested here. This advantage is bought at the expense of reducing each observation to a single binary code, according to whether it is above or below some hypothesized median value. Such information loss can lead to loss of statistical power, and wide confidence intervals.

Wilcoxon's signed rank test is also a permutation test, applied to the signed ranks. The results of the previous section were carried out using the same routine as for this section, applied to the signed ranks, with test statistic equal to the mean signed rank (or equivalently, the sum of the signed ranks, or the sum of the positive signed ranks), to test $H_0 : M = 0$, and to set a permutation interval for *M*. Note that the signed ranks change as *M* changes, so they must be re-evaluated at each step as the search for the limits of the permutation interval progresses.

Advantages of permutation tests and intervals:

1. The method is exact (i.e., same result obtained each time it is run on the same data).
2. No specific distribution is assumed for the data.
3. The analytic distribution of the test statistic is not required.

Disadvantages:

1. Each permutation of the data must be equally likely. (If they are not, but the probabilities of occurrence are known, permutations can be sampled with appropriate probability.)
2. The method is difficult or impossible to apply in complex examples.

2.4 Randomization test and randomization interval

Computer algorithms to generate all possible permutations are generally non-trivial to program. Further, it is often impractical time-wise to generate every possible permutation when carrying out a permutation test. Our definition of a randomization test is a permutation test in which a random subset of the possible permutations is generated. The proportion that yields a test statistic at least as extreme as the real sample is an estimate of the *p*-value. This definition is not standard, but is a convenient distinction to make; many authors use the terms permutation test and randomization test interchangeably. Algorithms to generate permutations at random are generally trivial. In the one-sample case, random permutations can be generated by listing the set of values $| x_i - \mu |$ (or $| x_i - M |$), and randomly assigning a sign to each value (P(+)=P(–)=0.5). The preferred test statistic is then evaluated, and examined to see if it is at least as extreme as that for the real data.

For the above example, and using 999 randomizations and then including the original dataset as the 1000th, we obtained a *p*-value of 0.015. This compares well with the exact *p*-value of 0.012 when all 4096 permutations are enumerated. This result may be illustrated more effectively by plotting a histogram, in which intervals for $\bar{x}$ are defined along the *x*-axis and the number of permutations for which $\bar{x}$ falls in a given interval determines the height of the corresponding bar. In Fig. 2.4.1, we show the randomization distribution for this example, and shade the bar that contains the observed sample mean of 0.074. (Note that we now get different estimates of the *p*-value if we double the one-tailed estimate or sum both tail areas; doubling the one-tailed estimate has lower precision.) Why is the distribution of Fig. 2.4.1 bimodal?

Table 2.4.1 shows an R program for performing this test and plotting a histogram.

Fig. 2.4.1. Histogram of sample means obtained from 999 randomizations of the pine marten data, plus the observed mean. The observed mean of 0.074 corresponds to one of the most

extreme permutations.

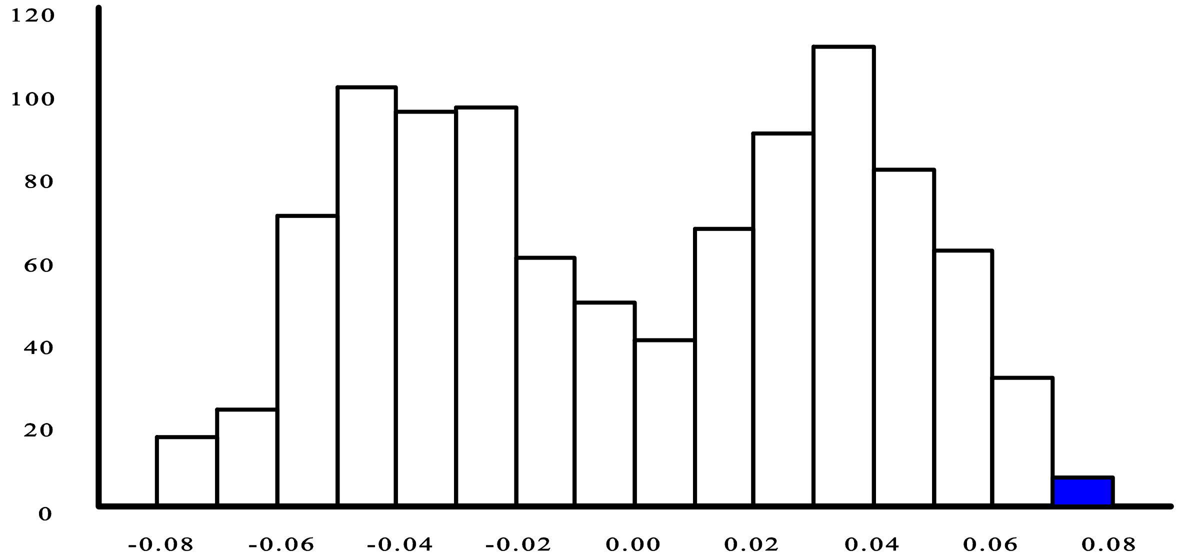


Table 2.4.1. R code for carrying out a randomization test of the null hypothesis of no selection or avoidance of deciduous woodland by pine martens against a two-tailed alternative. (Note: what is missing from this code?)

```
perm <- function (data,signs,nrand)
{
  rmean <-  numeric(nrand)
  n <- length(data)
  for (i in 1:nrand) {
    rsigns <-  sample(signs,n,replace=TRUE)
    rdata <-  data * rsigns
    rmean[i] <-  mean(rdata)
  }
  return(rmean)
}

data <- c(0.13, 0.01, 0.01,0.42, 0.02,0.01,0.09,0.03,0.04,0.06,
          0.12,0.03)
meandata <- mean(data)
signs <- c(-1,1)
nrand <- 999

rmean <- perm(data,signs,nrand)

rmean[nrand+1] <- meandata
rabsmean <-  abs(rmean)

meandata1 <- meandata-1E-10
extreme <- ifelse(rabsmean>meandata1,1,0)
p <-  sum(extreme)/length(rmean)
p

hist(rmean,br=20,xlim=c(-0.1,0.1))
```

Obtaining the exact permutation interval is far more computer-intensive than carrying out a single permutation test. Hence there is a need for efficient randomization methods for approximating the permutation interval. Robbins-Monro search provides a fully efficient method.

Suppose we seek a 95% confidence interval for the mean index value in our example. By assuming that index values were normally distributed, we earlier obtained a confidence interval of (-0.002, 0.150). We will use these limits as the start points for searches for better confidence limits, using randomization tests and Robbins-Monro search.

Consider the upper limit μ_u, and set up $H_0 : \mu = \mu_u$ against $H_1 : \mu < \mu_u$. At step *j*, denote the current estimate of μ_u by U_j. Subtract U_j from each observation, then generate a permutation at random from the resulting values. If $\bar{x}$ is the mean of the original data, and $\bar{y}_j = \bar{x}_j - U_j$ is the mean of values in the permutation generated at step *j*, then update the estimate of μ_u as follows:

$$U_{j+1} = \begin{cases} U_j - c\alpha / j\,, & \text{if } \bar{x}_j > \bar{x} \\ U_j + c(1-\alpha)/ j\,, & \text{if } \bar{x}_j \le \bar{x} \end{cases}$$

where $\alpha = 0.025$ and *c* is a step length constant.

This search oscillates widely for small *j*. Hence we start the search from a reasonable approximation to the upper confidence limit, 0.150 in the example, and arbitrarily start at, say, *j*=40 with U_{40}=0.150. The number of steps might vary from a few hundred to many thousand, depending on the precision required and the computing resources available. The optimal value of the step length constant depends on the true distribution of the test statistic, but performance of the method is generally good if it is set to a generic value, equal to double the optimal value for the normal distribution. It is calculated at step *j* as $c = k(U_j - \bar{x})$ where

$$k = \frac{2\sqrt{2\pi}}{z_\alpha \exp(-z_\alpha^2 / 2)}$$

and $z_\alpha = 1.96$ for $\alpha = 0.025$. Under very general conditions, U_j converges to μ_u as $j \to \infty$.

The lower limit is found similarly. An R program for calculating limits by Robbins-Monro search is given in Table 2.4.2.

Table 2.4.2. R code for calculating a 95% randomization interval for the habitat utilization index for deciduous woodland use by pine martens.

```
# Data are assumed to be in the vector data

rmean <- lows <- highs <- stepno <- NULL

# Calculate sample size and test statistic = sample mean
n <- length(data)
meandata <- mean(data)
for (i in 1:39) {
  rdata <- sample(data,n,replace=TRUE)
  rmean[i] <- mean(rdata)
}

# smallest and largest values from 39 give crude starting
# values for the lower and upper limits respectively
rmean <- sort(rmean)
loest <- rmean[1]
hiest <- rmean[39]
lows[1] <- loest
highs[1] <- hiest
stepno[1] <- 1
signs <- c(-1,1)

# now start Robbins-Monro search at step 40; 1000 steps in all
for (i in 40:1000) {
  dlo <- data-loest
  dhi <- data-hiest

  # lower limit first.
  rsigns <- sample(signs,n,replace=TRUE)
  rmean <- mean(dlo*rsigns)
  steplength <- 16*(meandata-loest)

  loest <- ifelse(rmean+loest<meandata,
              loest+steplength*0.025/(i-1),
              loest-steplength*0.975/(i-1))

  lows[i-38] <- loest

  # now upper limit
  rsigns <- sample(signs,n,replace=TRUE)
  rmean <- mean(dhi*rsigns)
  steplength <- 16*(hiest-meandata)

  hiest <- ifelse(rmean+hiest>meandata,
              hiest-steplength*0.025/(i-1),
              hiest+steplength*0.975/(i-1))

  highs[i-38] <- hiest
  stepno[i-38] <- i-38
}

loest
hiest
plot(lows,stepno)
plot(highs,stepno)
```

Randomization tests and intervals have the same advantages and disadvantages as permutation tests and intervals, except that the methods are easier to program, and are subject to Monte Carlo variability, so that they are exact only as the number of randomizations tends to infinity.

2.5 Monte Carlo test

In a Monte Carlo test, we set up a null hypothesis, then simulate a data set and calculate the test statistic, assuming that hypothesis is true. We repeat this a large number of times (traditionally 99, when simulations were done by hand, but preferably of the order of 999), add the test statistic evaluated using the real sample, then observe how extreme that value is in the ordered list. If it is among the 5% of most extreme values, then we have $p < 0.05$. Thus the randomization test is a special case of the Monte Carlo test, in which the simulated data sets are generated by selecting permutations of the data at random. In the one-sample case, if we have $H_0 : \mu = \mu_0$, then we obtain the random permutations by randomizing the signs of $x_i - \mu_0$, $i = 1,...,n$.

Consider again our example. We could carry out a Monte Carlo test as follows. Assume that the habitat utilization index values x follow a $N(\mu, \sigma^2)$ distribution, and that σ^2 is known. (In practice, we might estimate it by s^2, and proceed as if it was the true value.) As with permutation tests, we have considerable flexibility in choice of test statistic. Suppose we use the sample mean, $\bar{x}$. If we want to test $H_0 : \mu = 0$ (i.e., no preference or avoidance of deciduous woodland), we generate a set of $n = 12$ observations from $N(0, \sigma^2)$, and calculate the mean of this sample. We repeat this step, generating b data sets of size 12, and calculating the mean of each. (We can reduce the computation by noting that, given our assumptions and H_0, $\bar{x} \sim N(0, \sigma^2 / 12)$, so that we can generate b deviates directly from this distribution.) We now add the mean of the real sample, giving $b + 1$ values in total. If the alternative hypothesis is two-tailed, we count the number of values that are $\geq \bar{x}$ or $\leq -\bar{x}$. If this number is r, then the p-value is $\frac{r}{b+1}$. See Table 2.5.1.

Table 2.5.1. Monte Carlo test of whether the habitat utilization index for pine martens in deciduous woodland is zero.

```
# Data are assumed to be in the vector data.
meandata <- mean(data)
semean <- sd(data)/sqrt((length(data)))

rmean <- rnorm(999,0,semean)

rmean[1000] <- meandata

less <- ifelse(rmean<=meandata,1,0)
more <- ifelse(rmean>=meandata,1,0)

nless <- sum(less)
nmore <- sum(more)
if(nless<nmore) {nmore <- nless}

# Assume alternative hypothesis is two-tailed:
p <- nmore*2/1000

# Print p-value:
p
```

Advantages of Monte Carlo tests:

1. The data may be assumed to follow any specified distribution. It is merely necessary to fit that distribution, and generate new data from it. Randomization tests may be considered as a special case of Monte Carlo tests.
2. The analytic distribution of the test statistic is not required, so the method can be used with intractable test statistics.
3. The tests can often be made exact apart from Monte Carlo variation.

Disadvantages:

1. If allowance is not made for estimating nuisance parameters, the method is approximate.
2. The method does not allow interval estimation, although bootstrap confidence intervals may be regarded as an extension of Monte Carlo tests to interval estimation.
3. If the test is applied twice to a single data set, it will generally generate different *p*-values. The method can only be asymptotically exact as $b \to \infty$.

2.6 Nonparametric bootstrap

Permutation and randomization methods, and computer-intensive methods more generally, generate multiple samples (often called **resamples**), each of which is assumed to have the same properties as the real sample. Another approach, the bootstrap, is widely used for estimating variances and confidence intervals.

Consider our example data:

$\bar{x}$

0.13, -0.01, -0.01, 0.42, -0.02, 0.01, 0.09, 0.03, 0.04, 0.06, 0.12, 0.03   0.074

Suppose we wish to estimate the variance of the sample mean, and to set a 95% confidence interval for the mean index value in the population. In the nonparametric bootstrap, we assume that observations are independently and identically distributed. We approximate the true distribution function of $\bar{x}$ by the empirical distribution function, and generate a new sample (i.e., a resample) from it. This is done most simply by sampling observations from the original sample at random and **with replacement** until the bootstrap resample is of the same size as the original sample. We now repeat this *b* times, where typically *b* = 400-1000:

Bootstrap
resample   Observations

```
    1        -0.01 0.42 0.12 0.09 0.13 0.01 -0.02 0.12 -0.01 -0.01 0.12 -0.02    0.078
    2        0.06 0.01 0.13 0.04 0.06 -0.01 0.03 -0.01 -0.02 0.03 0.01 -0.01     0.027
.....
    b        -0.01 0.04 -0.02 0.42 0.03 0.12 0.13 -0.02 0.42 0.09 0.01 0.06      0.106
```

We assume that the means of the bootstrap resamples represent the variability we would observe in the sample mean if we were to obtain many samples of size 12 from the population of pine martens. The sample variance of the bootstrap means is an estimate of the variance of $\bar{x}$, and approximate confidence intervals may be found using the percentile method as follows:

Order the bootstrap means from smallest to largest. Approximate 100(1-2α)% confidence limits are given by the *r*th and *s*th values in this list, where $r = \alpha(b+1)$ and $s = (1-\alpha)(b+1)$. For example if we generate *b* = 999 resamples, then approximate 95% confidence limits are given by the 25th smallest and 25th largest estimates from the ordered list.

In Table 2.6.1, we use R to obtain an estimated standard error for $\bar{x}$ and 95% percentile confidence limits for μ in the above example, using 999 nonparametric bootstrap resamples.

Table 2.6.1. R code for calculating a nonparametric bootstrap standard error and a 95% percentile confidence interval for the habitat utilization index for deciduous woodland use by pine martens. One could also consider using functions in the R packages `bootstrap` or `boot` to do this.

```
# Data are assumed to be in the vector data.
# Calculate sample size and sample mean.

meandata <- mean(data)
n <- length(data)
nboot <- 999
alpha <- 0.025       # i.e. 95% confidence interval
bootmean <- NULL

for (i in 1:nboot) {
  rdata <- sample(data,n,replace=TRUE)
  bootmean[i] <- mean(rdata)
}

bootse <- sd(bootmean)
nlo <- round((nboot+1)*alpha+0.49)
nhi <- round((nboot+1)*(1-alpha)+0.49)
bootmean <- sort(bootmean)
low <- bootmean[nlo]
high <- bootmean[nhi]

# print bootstrap s.e. and 95% percentile confidence limits
results <- c(bootse,low,high)
results
```

The **balanced bootstrap** has marginally better performance than the standard bootstrap. The *n* observations are listed *b* times, and the full list of *nb* values is randomly reordered. The first bootstrap resample comprises the first *n* observations from the reordered list, and so on. This ensures that each observation is used exactly *b* times in the *b* resamples.

Advantages of the nonparametric bootstrap:

1. It is a simple technique to apply.
2. It is a general and robust (when compared with general analytic methods) method of setting confidence limits.

Disadvantages:

1. It assumes that the observations are independently and identically distributed (i.i.d.).
2. In complex applications, it is often unclear what the unit for resampling should be. If no appropriate choice exists, the performance of the method may be very poor.
3. The method is generally only asymptotically exact as both *b* and *n* tend to infinity.

2.7 Parametric bootstrap

For the parametric bootstrap, instead of resampling from the observations, we first fit a parametric model to the data. Resamples are then simulated from this fitted model. These resamples are then analysed in the same way as for the nonparametric bootstrap.

In our example, we obtained approximately $\bar{x} = 0.0742$ and $s^2 = 0.0143$. If we choose to model the distribution of *x* by the normal distribution, the fitted distribution is *N*(0.0742, 0.0143). To apply the parametric bootstrap, we therefore generate 12 values from this distribution, and calculate their mean. We repeat this process *b* times, then proceed as for the nonparametric bootstrap. Note that, in this case, as for the example of a Monte Carlo test, we can take a short cut, since the fitted model implies that $\bar{x} \sim N(\hat{\mu}, \hat{\sigma}^2 / n)$, where $\hat{\mu} = \bar{x}$, $\hat{\sigma}^2 = s^2$, and $n = 12$. We can therefore generate the *b* bootstrapped values of the mean directly from this distribution. This method is essentially an extension of Monte Carlo testing to variance and confidence interval estimation. An R program for the method is given in Table 2.7.1.

If we knew that the data were normally distributed, we would not need to bootstrap. Apart from Monte Carlo variation, the above method yields the interval $\bar{x} \pm 1.96 s / \sqrt{n} = (0.0065, 0.1419)$, which is the analytic interval obtained assuming σ^2 is known. To obtain the interval $\bar{x} \pm t_{11}(0.025) s / \sqrt{n} = (-0.002, 0.150)$ apart from Monte Carlo variation, we would generate *b* deviates t_j from Student's *t* distribution with 11df, and transform them to $\bar{x}_j = t_j \sqrt{0.0143/12} + 0.0742$. Again, we can do better analytically. The value of the parametric bootstrap is that we can readily apply it for distributions other than the normal, for which we may not have analytic results, or for which we must make crude approximations to obtain analytic results.

Table 2.7.1. R code for calculating a parametric bootstrap standard error and a 95% percentile confidence interval for the habitat utilization index for deciduous woodland use by pine martens.

```
# Data are assumed to be in the vector data.

meandata <- mean(data)
semean <- sd(data)/sqrt((length(data)))
nboot <- 999
alpha <- 0.025      # i.e. 95% confidence

bootmean <- rnorm(nboot,meandata,semean)

bootse <- sd(bootmean)
nlo <- round((nboot+1)*alpha+0.49)
nhi <- round((nboot+1)*(1-alpha)+0.49)
bootmean <- sort(bootmean)
low <- bootmean[nlo]
high <- bootmean[nhi]

# print bootstrap s.e. and 95% percentile confidence limits
results <- c(bootse,low,high)
results
```

Advantages of the parametric bootstrap:

1. It is a general and robust (when compared with general analytic methods) method of setting confidence limits.
2. Unlike the nonparametric bootstrap, observations need not be i.i.d.

Disadvantages:

1. The method is generally only asymptotically exact as both *b* and *n* → ∞.
2. A parametric model must be assumed.

2.8 Test-inversion bootstrap intervals

There are several bootstrap methods in the literature that yield better confidence intervals than the percentile method. For most practical purposes, the improvement does not warrant their additional complexity. For single parameter problems, undoubtedly the best method is the test-inversion method, because it converges to the exact interval as *b* tends to infinity, for any sample size *n*, and for any continuous univariate distribution. If the interval endpoints are obtained by Robbins-Monro search, it also attains maximum precision for a given number of bootstrap replicates, if the step length constant is known. As we said earlier, the term "test-inversion" refers to the fact that the confidence interval is found as the range of values for the parameter that are not rejected using hypothesis testing at the corresponding significance level.

The method is applied exactly as described for randomization intervals, except that, instead of generating a randomization at step *j*, a parametric bootstrap resample is generated, in which the assumed distribution is fitted by setting the value of the (single) parameter to the current estimate of the interval endpoint.

The method also compares favourably with most other approaches in the presence of nuisance parameters. The nuisance parameters are fixed at their estimated values, and a Robbins-Monro search conducted for the interval endpoints for the parameter of interest. In this circumstance, it no longer converges to an exact interval as *b* tends to infinity, unless sample size *n* also tends to infinity (and consistent estimators are used).

2.9 Jackknife

Another method of generating resamples is to omit each observation in turn, thus creating *b* = *n* resamples. Each resample is analysed as if it was the real sample. Using our example, and assuming that we want to draw inferences about μ, we obtain estimates $\bar{x}_{(j)}$, *j* = 1, ..., *n*, where (*j*) indicates that observation *j* was omitted from the sample. Pseudovalues $\bar{x}^{(j)}$ are then calculated as

$$\bar{x}^{(j)} = n\bar{x} - (n-1)\bar{x}_{(j)} \ , \ j = 1, ..., n$$

The variance of *x* is then estimated by the sample variance of the pseudovalues, so that

$$\hat{V}_j(\bar{x}) = \frac{\sum_{j=1}^{n}(\bar{x}^{(j)} - \bar{x}')^2}{n(n-1)}$$

where $\bar{x}' = \sum_{j=1}^{n} \bar{x}^{(j)} / n$.

Note that in this particular example, $\bar{x}^{(j)} = x_j$, so that the above variance is simply s^2 / n , where

s^2 is the sample variance of the observations.

To estimate confidence limits, the pseudovalues are usually assumed to be normally distributed, so that an approximate 100(1-α)% confidence interval is given by

$$\bar{x}' \pm t_{n+1}(\alpha/2)\sqrt{\hat{V}_j(\bar{x})}$$

Advantages of the jackknife:

1. The jackknife is easily implemented.
2. The jackknife is balanced, in that each observation is used the same number of times (n -1).

Disadvantages:

1. Pseudovalues are assumed to be i.i.d., and, to calculate confidence intervals, they are also assumed to be normally distributed. (In general, the pseudovalues are correlated, and may be markedly non-normal.)
2. There is no control over the number of resamples: $b = n$.

3. Two-sample methods

For paired data, observations can be reduced to a single sample by taking the differences between pairs. The one sample methods described above may then be applied. For two independent samples, we summarize the modifications required to the above methods. We again use data on pine martens, this time fictitious, to illustrate techniques:

The percentages of time spent foraging in deciduous woodland by individual pine martens during a radio-tracking study were as follows.

Males	10	4	85	5	7	92	0	13
Females	23	41	35	46	51	26		

We wish to draw inferences about the relative use of this habitat by the two sexes.

The conventional analytic method for comparing the means of two independent samples is a t-test, in which a pooled estimate of sample variance, obtained as an average of the two estimates, weighted by their degrees of freedom, is found. The test is valid provided the variance is the same in each sample, in addition to the assumption that data in both samples are normally and independently distributed. In this case, there seems a clear difference in variance between the groups, and a transformation of the percentages would be advisable if a normal distribution is assumed. However, for illustration purposes, we apply the standard test regardless.

We have $n_m = 8$ $\quad \bar{x}_m = 27.00$ $\quad s_m^2 = 10216/7 = 1459.4$

$n_f = 6$ $\quad \bar{x}_f = 37.00$ $\quad s_f^2 = 614/5 = 122.8$

The pooled estimate of variance is therefore

$$s_p^2 = \frac{(n_m - 1)\, s_m^2 + (n_f - 1)\, s_f^2}{n_m + n_f - 2} = \frac{10216 + 614}{12} = 902.5$$

To test $H_0 : \mu_m = \mu_f$ against the two-sided alternative $H_1 : \mu_m \_ \mu_f$, we have

$$\frac{(\bar{x}_m - \bar{x}_f) - (\mu_m - \mu_f)}{\sqrt{s_p^2\left(\frac{1}{n_m} + \frac{1}{n_f}\right)}} = \frac{(27 - 37) - 0}{\sqrt{902.5\,(1/8 + 1/6)}} = -0.616$$

Comparing this with Student's *t* distribution with $n_m + n_f - 2 = 12$df, it is clearly non-significant.

We can obtain a nominal 95% confidence interval for $\mu_m - \mu_f$ as:

$$\frac{(\bar{x}_m - \bar{x}_f) - (\mu_m - \mu_f)}{\sqrt{s_p^2\left(\frac{1}{n_m} + \frac{1}{n_f}\right)}} = \frac{(27 - 37) - (\mu_m - \mu_f)}{\sqrt{902.5\,(1/8 + 1/6)}} = \pm\, t_{12}(0.025) = \pm\, 2.179$$

which gives the interval (-45.4, 25.4).

However, if we test $H_0 : \sigma_m^2 = \sigma_f^2$ against a two-sided alternative, we obtain

$$F = \frac{s_m^2 / \sigma_m^2}{s_f^2 / \sigma_f^2} = 1459.4 / 122.8 = 11.88$$

This is a value from the *F* distribution with 8 and 6 df if H_0 is true. The value is significant at the 1% level, suggesting that we cannot reasonably assume that the variances of the two samples are equal.

Approximate tests and confidence intervals are available when the variances differ. The test statistic

$$\frac{(\bar{x}_m - \bar{x}_f) - (\mu_m - \mu_f)}{\sqrt{\frac{s_m^2}{n_m} + \frac{s_f^2}{n_f}}}$$

is approximately distributed as Student's *t* with df v given by

$$\frac{1}{v} = \frac{s_m^4}{k^2 n_m^2 (n_m - 1)} + \frac{s_f^4}{k^2 n_f^2 (n_f - 1)}, \text{ where } k = \frac{s_m^2}{n_m} + \frac{s_f^2}{n_f}$$

(Both this and the equal-variance version of the two-sample t-test are available in R via the `t.test` function: there's a `var.equal` function argument with default `FALSE`.)

For our example, a test of $H_0 : \mu_m = \mu_f$ yields $t = 0.702$ with df v = 8.5. In this case, conclusions are clearly unaffected when we allow for the variances to differ. The confidence interval is also only slightly affected. Approximating v by 8, we obtain the interval (-42.8, 22.8). Generally, provided $n_m \_ n_f$, we lose little by assuming that the variances are equal.

To carry out a permutation test of $H_0 : \mu_m = \mu_f$, we generate every possible combination of n_m observations from $n = n_m + n_f$, evaluate our test statistic for each combination, and determine how many are at least as extreme as the observed split between males and females. We can use $\bar{x}_m - \bar{x}_f$ as the test statistic. However, the ordering of the combinations, and hence the test, is unaffected if we take instead $\sum x_f$, the sum of observations in the smaller sample, and as this statistic is easier to evaluate, we use it here. In our example, $\sum x_f = 222$. What combinations are at least as extreme as this? $\bar{x}_m < \bar{x}_f$, so if we had $H_1 : \mu_f > \mu_m$, then any combination for which $\sum x_f \geq 222$ would be at least as extreme. But what if $H_1 : \mu_f \neq \mu_m$? We have $\bar{x}_m - \bar{x}_f = -10$. Values at least as extreme are $\bar{x}_m - \bar{x}_f \leq -10$ and $\bar{x}_m - \bar{x}_f \geq 10$. For every combination,

$$\sum x_f + \sum x_m = 222 + 216 = 438, \qquad (1)$$

because randomly reassigning the 14 observations to the two samples does not affect their overall sum. If $\bar{x}_m - \bar{x}_f = 10$, then multiplying both sides by 24 yields

$$3\sum x_m - 4\sum x_f = 240 \qquad (2)$$

3(1) - (2) yields: $7\sum x_f = 1074$

Thus values in the other tail at least as extreme as $\sum x_f = 222$ are $\sum x_f \leq 153.4$. We can now enumerate combinations at least as extreme as the observed one, assuming that male and female pine martens do not differ in their preference for deciduous woodland:

Samples of size 6 (from 14)	$\sum x_f$
92, 85, 51, 46, 41, 35	350
92, 85, 51, 46, 41, 26	341
....	
13, 10, 7, 5, 4, 0	39

There are $\binom{14}{6}$ = 3003 samples of size 6 in total. Clearly, many of these are at least as extreme as the observed sample. A computer algorithm is required to determine the actual number.

In Table 3.1, an R program is given that carries out a randomization test for comparing two means. Note the similarities with, and differences from, the program in Table 2.4.1.

Table 3.1. R code for carrying out a randomization test of the null hypothesis of no difference in utilization of deciduous woodland by male and female pine martens against a two-tailed alternative.

```
# Two samples are in data1 and data2

meandata1 <- mean(data1)
n1 <- length(data1)
data <- c(data1,data2)
rmean <- NULL

for (i in 1:1000) {
  rdata1 <- sample(data,n1,replace=FALSE)
  rmean[i] <- mean(rdata1)
}

more <- NULL
less <- NULL

more <- ifelse(rmean>=meandata1,1,0)
less <- ifelse(rmean<=meandata1,1,0)

nmore <- sum(more)
nless <- sum(less)
if(nless<nmore) {nmore <- nless}

# Assume alternative hypothesis is two-tailed:
p <- 2*nmore/1000

# Print p-value
p
```

Choice of an appropriate test statistic is important. In the above, the difference in sample means (or equivalently, the sum of observations from one of the samples) is useful for testing whether the locations of the two distributions differ. To test whether the variances differ, a more useful test statistic would be the ratio of sample variances. This is closely comparable to the *F*-test of whether the variances differ, but without the requirement that observations are normally distributed.

The permutation test assumes that, under H_0, the observations are interchangeable. Hence they are assumed to be i.i.d. In particular, they are assumed all to have the same mean and variance. The null hypothesis specifies that the means are the same, but we would prefer, for these data at least, to avoid the assumption that the variances are the same. We can do this by pooling the two samples, and replacing observations by their ranks in the combined sample. The above test statistic then becomes the sum of the ranks of observations from the smaller sample. In fact, this permutation test is then equivalent to both the Mann-Whitney test and Wilcoxon's rank sum test.

If we have $H_0 : \mu_f = \mu_m + d$, then we subtract *d* from every observation in the second sample, giving the new set of observations $x_{m1}, \ldots, x_{mn_m}, x_{f1} - d, \ldots, x_{fn_f} - d$, and the test proceeds as before. Thus a 100(1-α)% permutation interval for the difference $\mu_f - \mu_m$ is found by determining the set of values d for which H_0 is not rejected at the 100α% level (two-tailed test). Randomization tests and intervals follow in the obvious way.

Monte Carlo tests extend in the obvious way to two samples. Distributions are specified for both samples. If the null hypothesis is that the two distributions have the same mean, this too is specified, and new samples are generated under this assumption. A test statistic such as $\bar{x}_m - \bar{x}_f$ is then evaluated for the real samples and each simulated set of samples, and the significance level evaluated as for the one sample case.

A nonparametric bootstrap resample is obtained by sampling observations with replacement from each sample independently, so that the two samples in the resample are again of sizes n_m and n_f. In total, b resamples are generated, and if we want to draw inferences about $\mu_m - \mu_f$, we evaluate $\bar{x}_m - \bar{x}_f$ for each resample, and hence estimate the variance of $\bar{x}_m - \bar{x}_f$ and a confidence interval for $\mu_m - \mu_f$ in the usual way. In the case of the parametric bootstrap, distributions are specified for both samples, as in the Monte Carlo test, but unlike that test, resamples are generated from these distributions without any constraint on the parameters imposed by a null hypothesis. The resamples are analysed as for the nonparametric bootstrap.

4. Other standard analyses

We have used one sample methods to illustrate a range of computer-intensive techniques, and to compare them with standard methods. We also briefly examined two sample methods, as a further illustration of how the techniques are applied. Here, we indicate briefly how other standard analyses might be done in a computer-intensive way. The major value of computer-intensive methods is that they make many non-standard analyses possible and practical, so we do not dwell at length in this course on standard analyses.

4.1 One-way analysis of variance

Suppose there are m groups, with n_j observations in group j, and $\sum n_j = n$. Perhaps the simplest computer-intensive analysis to apply is the randomization test. A randomization is generated by pooling all n observations, then randomly assigning them to m groups, with n_j observations in group j. If we wish to test $H_0 : \mu_1 = ... = \mu_m$, then a suitable test statistic is the between groups sum of squares, $\sum T_j^2 / n_j - T^2 / n$, where T_j denotes the sum of observations in group j, and T denotes the overall sum of observations. Relative to the random permutations, we expect the real data to generate a large value for this statistic if the null hypothesis is false.

4.2 Two-way analysis of variance

Suppose we have a set of treatments, and the experiment is arranged in blocks, with a single observation on each treatment within each block. If the null hypothesis is that the treatments do not differ, the observations within a block are exchangeable. Hence we can apply a randomization test by randomly permuting observations *within* blocks. That is, an observation is retained in its block in every randomization, but it is randomly reassigned to treatment. Again, the treatment sum of squares $\sum T_j^2 / n_j - T^2 / n$ can be used as a test statistic.

The methods readily extend to more complex analyses of variance. See Manly (1997).

4.3 Contingency tables

Consider the two-way table of counts:

		Ability in maths			
		Above average	Average	Below average	TOTAL
Ability in music	Above average	10	7	5	22
	Average	8	15	7	30
	Below average	3	6	9	18
	TOTAL	21	28	21	70

We wish to assess whether there is any association between ability in maths and ability in music. The normal contingency table test conditions on the marginal totals. Random permutations of these data can be generated by reassigning the 70 people randomly to the nine cells. If we condition on only those permutations that give rise to the above marginals, then in principle, every possible permutation with this property can be identified. We could use the standard contingency table goodness-of-fit statistic, $\chi^2 = \Sigma[(O - E)^2 / E]$, as our test statistic. We calculate the proportion of permutations with the above marginal totals that yield a value of χ^2 at least as large as that for the real data, to obtain the significance level for testing the null hypothesis of no association. This test is exact, whereas the usual test, assuming that the above statistic has a χ^2 distribution with 4 df, is approximate. For a two by two table, the test is termed Fisher's exact test. In practice, for larger tables, we would carry out a randomization test. The 70 people would be randomly assigned one at a time to the nine cells. This can be done as follows. Select a number at random between 1 and 70. If the number is at most 22, assign the individual to row one. If between 23 and 52, assign to row 2, and if >52, assign to row 3. Select a second number between 1 and 70, and assign the individual to a column in the same way. Suppose the individual is assigned to row 2 and column 1. We now reduce the corresponding marginal totals to 29 and 20 respectively, and the overall total to 69. We now select two numbers at random between 1 and 69, to assign the second individual, and so on.

4.4 Multiple regression

In multiple regression, we condition on the covariates. If we wished to carry out a regression without assuming that errors are normally distributed, a simple method is to bootstrap the units. A unit is the observation y_i, along with the associated covariate values, $\underline{x}_i$. In practice, this method works well, provided the number of units is reasonably large (say 15 or more). However, because the covariate values are being resampled along with the observations, this analysis fails to condition on the covariates. A simple solution is to fit the model (usually by least squares), calculate the predictions $\hat{y}_i$ and the residuals $r_i = y_i - \hat{y}_i$, $i = 1, ..., n$, then resample those residuals. That is, if there are n observations, a sample of size n is drawn randomly and with replacement from the n residuals. The observations in the bootstrap sample are then found by adding the bootstrapped residual r_i^* say to $\hat{y}_i$, to obtain the bootstrap observation y_i^*, $i = 1, ..., n$. b bootstrap resamples are generated in this way, and inference proceeds as before. That is, any parameter or prediction of interest is estimated from each bootstrap resample, and bootstrap variances and confidence intervals calculated in the usual way. This method assumes that the residuals are independently and identically distributed (which is only reasonable if $n \ll p$, where p is number of parameters in the model), but does not assume that they are normally distributed.

A parametric bootstrap resample could be obtained by generating bootstrap residual r_i^* from a $N(0\,,\; s^2\,)$ distribution, where s^2 is the residual mean square. If n is small, better results are obtained by generating a value t_i^* from Student's t distribution with n - p df, where p is the number of parameters estimated. The bootstrap residual is then $r_i^* \;=\; t_i^* \, x \, s$.

Randomization methods may be used specifically to test the slope(s) in regression. Under H_0: β=0 (where β is a vector if there is more than one covariate), there is no link between the observations y and their covariates. Thus we can randomly reallocate the observations to the (sets of) covariates to generate randomizations. The slope(s) is/are estimated as in conventional regression, and we observe the proportion(s) of slopes that is/are at least as far from zero as the estimated slope(s) calculated from the real data. Note that each randomization has the same sample mean. Since the slope estimate(s) is/are independent of the sample mean, it does not matter that the randomizations ignore the component of variance due to the sample mean.

Question: How would you apply the randomization test under H_0: β=β_0≠0?

5. Extensions of computer-intensive methods

The primary advantage of computer-intensive methods is their flexibility. We illustrate this through a series of examples of ways in which they may be extended.

5.1 Probability Integral Transform Resampling

Probability Integral Transform Resampling (PITR) is a method for bootstrapping observations that are independently but not necessarily identically distributed:

Consider observation y_i with cumulative density function (cdf) $F_i(\,y_i)$

1. Transform y_i to $u_i = \hat{F}_i(\,y_i)$.

2. Resample from the u's (or randomly permute the u's) - justified because $F_i(y_i) \sim$ U(0,1) for all i. i.e., if F_i were known, the u's would be i.i.d.

3. Reassign the u's to observations by assigning the i^{th} u in the resampled (or permuted) list to the i^{th} observation y_i in the original list. If u_j is assigned to observation y_i, bootstrap observation i is calculated as $\hat{F}_i^{-1}(\,u_j)$.

Observations must be independently distributed, but need not be identically distributed.

5.2 Bootstrapping generalized linear models

In multiple regression, we have a model of the form: $y_i = \alpha + \sum_j \beta_j\ x_{ij} + e_i$

where the errors are assumed to be normally distributed with mean zero and constant variance. The systematic component of this model, $\alpha + \sum_j \beta_j\ x_{ij}$, is termed the linear predictor. It can be expressed as $\underline{\eta} = X\underline{\beta}$, where

$$X = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots \\ 1 & x_{21} & x_{22} & \cdots \\ & & \cdots & \\ 1 & x_{n1} & x_{n2} & \cdots \end{pmatrix}$$

and $$\underline{\beta} = \begin{pmatrix} \alpha \\ \beta_1 \\ \beta_2 \\ \cdots \end{pmatrix}$$

Hence the vector $\underline{\eta}$ comprises the expectations of the observations. For generalized linear models, we have $y_i = g^{-1}(\ \alpha + \sum_j \beta_j\ x_{ij}\ ) + e_i$

or equivalently, $\underline{y} = g^{-1}(\underline{\eta}) + \underline{e}$, where $g(.)$ is the **link function**, so named because it provides the link between the random and systematic components. We also allow the errors to have distributions other than the normal, and allow their variance to be a function of the expectations of the observations. Errors are still assumed to be independently distributed with mean zero.

The nonparametric bootstrap assumes that the bootstrap units are i.i.d. The distributions of the errors are not now identical, which rules out resampling residuals, or observations together with the associated covariates. The parametric bootstrap is still applicable. We specify an error distribution, fit it using the data, then generate bootstrap residuals from the fitted distribution. In practice, it is simpler and more direct to generate bootstrap observations directly. For example in Poisson regression, the error distribution is specified as Poisson, and the link function is log (so that $g^{-1}(.)$ is exponential). It is simpler to obtain the predictions $\hat{y}_i$ from the fitted model, then generate bootstrap observation y_i^* as a deviate from the Poisson distribution with rate $\hat{y}_i$.

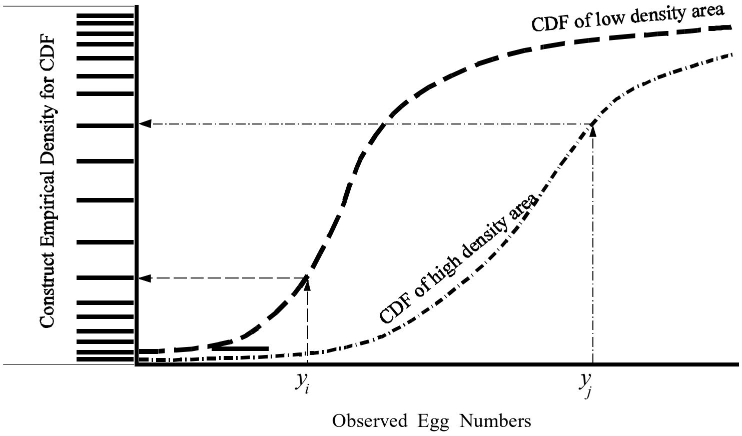
In Poisson regression, it is assumed that $V(y_i) = E(y_i)$. To generalize the approach, a **dispersion parameter** φ can be estimated, where $V(y_i) = \varphi\ E(y_i)$. The parametric bootstrap cannot now be applied, unless we can find a way of generating overdispersed Poisson deviates. PITR (Section 5.1) combines features of the nonparametric and parametric bootstraps, to allow generation of bootstrap resamples when errors are not identically distributed, and in a way that preserves overdispersion:

We obtain the predicted value $\hat{y}_i$ from the fitted model, corresponding to observation y_i, and hence estimate that the distribution of y_i is Poisson with mean $\hat{y}_i$. This determines the estimated cdf of the observation Y_i, which we evaluate at $Y_i = y_i$, to give u_i. In the absence of overdispersion, the resulting values are approximately uniformly distributed on (0,1). (The approximation arises because we estimate $E(y_i)$ by $\hat{y}_i$.) We now resample the u_i, reassign the resampled u's to the predicted values, and back-transform to generate the bootstrap sample of observations. In the presence of overdispersion, the u_i's are not approximately uniform, but instead are more likely to be close to zero or one. The above procedure preserves the overdispersion in the bootstrap resamples.

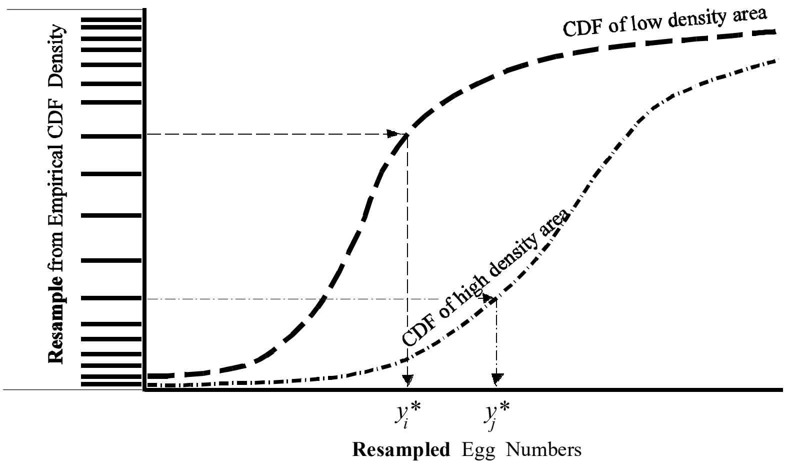
The procedure is illustrated in Fig. 5.2.1. In this example, a model was fitted to counts of mackerel eggs, as part of a stock assessment. Fig. 5.2.1(a) illustrates the process of constructing the empirical distribution of u_i values from the sampled egg numbers using the fitted model. (Two realisations of the fitted model are represented in the figure, the dashed line for a location of low egg density and the dot-dashed line for a location of high egg density.) Overdispersion is illustrated by the clustering of the u_i values of the sampled numbers towards 0 and 1. Fig. 5.2.1(b) illustrates the resampling process in which the u_i values are sampled at random and with replacement, and hence reassigned to the sampled locations and back-transformed to egg numbers, using the fitted model evaluated at the appropriate location.

Fig. 5.2.1 Diagrammatic representation of the resampling algorithm for generating overdispersed Poisson variates.

(a): Step 1. Construction of the empirical cumulative distribution function (cdf). Each observation *y* is transformed into an 'observed' cdf value using the Poisson distribution with mean equal to the GAM fitted value at that point. (Overdispersion results in clustering of the 'observed' cdf values towards 0 and 1, as indicated on the vertical axis.) The transformation is shown for two observations only.

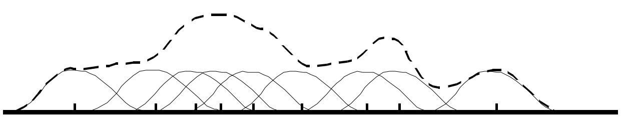


(b): Step 2. Construction of a new sample from the the empirical cdf. By permuting the 'observed' cdf values, resampled cdf values are generated for each of the sampled points. Each of these is then back-transformed using the Poisson distribution with mean equal to the GAM fitted value at that point to yield resampled *y*'s. The back-transformation is shown for the same two observations as in (a) and for the case in which permutation resulted in swapping the two associated cdf values.



5.3 The smoothed bootstrap

Especially when sample size *n* is small, the nonparametric bootstrap is limited because the same *n* observations are reused repeatedly. The smoothed bootstrap is sometimes used to overcome this problem. Conceptually, we can smooth the observed sample, then resample from the smoothed density instead of the empirical density. Suppose we smooth the data using a kernel density, in which each observation is replaced by a normal density with standard deviation *h*:



If the data are $y_1, \dots, y_n$, then a Gaussian kernel density estimate is defined by

$$\hat{f}(t\,;\,h) = \frac{1}{nh} \sum_1^n \phi\left(\frac{t - y_i}{h}\right),$$

where $\phi(t)$ is the standard normal density $\exp(-t^2/2) \,/\, \sqrt{2\pi}$. The parameter *h* is the *window size*; the larger the value of *h*, the greater the degree of smoothing.

Now generate a nonparametric bootstrap sample $y_1^*, \dots, y_n^*$ say, by drawing *n* values at random and with replacement from the original sample. A smoothed bootstrap resample, $x_1^*, \dots, x_n^*$ say, is obtained by calculating

$$x_i^* = \bar{y}^* + (1 + h^2 / s^2)^{-0.5} (y_i^* - \bar{y}^* + h\,\varepsilon_i)$$

for *i* = 1, ..., *n*, where $\bar{y}^*$ is the mean of the y_i^*, s^2 is the sample variance of the observations y_i, and ε_i are random errors drawn from *N*(0, 1). The term $(1 + h^2 / s^2)^{-0.5}$ is a scaling factor, to ensure that the bootstrap sample has approximate variance s^2. (Smoothing increases the variance if the scale factor is not used.) If the scale factor was excluded, this procedure would merely add a random perturbation with standard deviation *h* to the bootstrapped observation. The nonparametric bootstrap is obtained when *h*=0, and the limit as *h* tends to infinity gives the parametric bootstrap, with

$$x_i^* = \bar{y}^* + s\,\varepsilon_i$$

5.4 Incorporating model selection uncertainty into inference

Although model selection is widely recognised as central to good inference, paradoxically, it has seldom been integrated fully into inference. For example, there are many methods in multiple regression for identifying an appropriate subset of covariates. Having identified them, subsequent inference is usually conditional on the selected model; that is, we assume that the model is correct. It is more defensible to recognise the uncertainty in model selection when quantifying the precision of an estimator. Under this philosophy, “model mis-specification bias” is not bias at all, but merely a component of the variance. The difficulty in incorporating model selection uncertainty into inference can be circumvented using the bootstrap. The model selection procedure is applied independently in each bootstrap resample, and inference is based on the resulting bootstrap estimates.