5

One- and two-sample tests

Most of the rest of this book describes applications of R for actual statistical analysis. The focus to some extent shifts from explanation of the syntax to description of the output and specific arguments to the relevant functions.

Some of the most basic statistical tests deal with comparing continuous data either between two groups or against an a priori stipulated value. This is the topic for this chapter.

Two functions are introduced here, namely `t.test` and `wilcox.test` for t tests and Wilcoxon tests, respectively. Both can be applied to one- and two-sample problems as well as paired data. Notice that the "two-sample Wilcoxon test" is the same as the one called the "Mann–Whitney test" in many textbooks.

5.1 One-sample t test

The t tests are based on an assumption that data come from the normal distribution. In the one-sample case we thus have data $x_1, \ldots, x_n$ assumed to be independent realizations of random variables with distribution $N(\mu, \sigma^2)$, which denotes the normal distribution with mean μ and variance σ^2, and we wish to test the *null hypothesis* that $\mu = \mu_0$. We can estimate the parameters μ and σ by the empirical mean $\bar{x}$ and standard

P. Dalgaard, *Introductory Statistics with R*,
DOI: 10.1007/978-0-387-79054-1_5, © Springer Science+Business Media, LLC 2008

deviation s, although we must realize that we could never pinpoint their values exactly.

The key concept is that of the *standard error of the mean*, or SEM. This describes the variation of the average of n random values with mean μ and variance σ^2. This value is

$$\text{SEM} = \sigma/\sqrt{n}$$

and means that if you were to repeat the entire experiment several times and calculate an average for each experiment, then these averages would follow a distribution that is narrower than that of the original distribution. The crucial point is that even based on a single sample, it is possible to calculate an empirical SEM as $s/\sqrt{n}$ using the empirical standard deviation of the sample. This value will tell us how far the observed mean may reasonably have strayed from its true value. For normally distributed data, the rule of thumb is that there is a 95% probability of staying within $\mu \pm 2\sigma$, so we would expect that if μ_0 were the true mean, then $\bar{x}$ should be within 2 SEMs of it. Formally, you calculate

$$t = \frac{\bar{x} - \mu_0}{\text{SEM}}$$

and see whether this falls within an *acceptance region* outside which t should fall with probability equal to a specified *significance level*. This is often chosen as 5%, in which case the acceptance region is almost, but not exactly, the interval from -2 to 2.

In small samples, it is necessary to correct for the fact that an empirical SEM is used and that the distribution of t therefore has somewhat "heavier tails" than the $N(0,1)$: Large deviations happen more frequently than in the normal distribution since they can result from normalizing with an SEM that is too small. The correct values for the acceptance region can be looked up as quantiles in the t distribution with $f = n - 1$ degrees of freedom.

If t falls outside the acceptance region, then we reject the null hypothesis at the chosen significance level. Alternatively (and equivalently), you can calculate the *p-value*, which is the probability of obtaining a value as numerically large as or larger than the observed t and reject the hypothesis if the p-value is less than the significance level.

Sometimes you have prior information on the direction of an effect; for instance, that all plausible mechanisms that would cause μ not to equal μ_0 would tend to make it bigger. In those cases, you may choose to reject the hypothesis only if t falls in the upper tail of the distribution. This is known as *testing against a one-sided alternative*. Since removing the lower tail from the rejection region effectively halves the significance level, a one-sided test at a given level will have a smaller cutoff point. Similarly, p-values

are calculated as the probability of a larger value than observed rather than a numerically larger one, effectively halving the p-value as long as the observed effect is in the stipulated direction. One-sided tests should be used with some care, preferably only when there is a clear statement of the intent to use them in the study protocol. Switching to a one-sided test to make an otherwise nonsignificant result significant could easily be regarded as dishonest.

Here is an example concerning daily energy intake in kJ for 11 women (Altman, 1991, p. 183). First, the values are placed in a data vector:

```
> daily.intake <- c(5260,5470,5640,6180,6390,6515,
+ 6805,7515,7515,8230,8770)
```

Let us first look at some simple summary statistics, even though these are hardly necessary for such a small data set:

```
> mean(daily.intake)
[1] 6753.636
> sd(daily.intake)
[1] 1142.123
> quantile(daily.intake)
  0%  25%  50%  75% 100%
5260 5910 6515 7515 8770
```

You might wish to investigate whether the women's energy intake deviates systematically from a recommended value of 7725 kJ. Assuming that data come from a normal distribution, the object is to test whether this distribution might have mean $\mu = 7725$. This is done with `t.test` as follows:

```
> t.test(daily.intake,mu=7725)

         One Sample t-test

data:  daily.intake
t = -2.8208, df = 10, p-value = 0.01814
alternative hypothesis: true mean is not equal to 7725
95 percent confidence interval:
 5986.348 7520.925
sample estimates:
mean of x
 6753.636
```

This is an example of the exact same type as used in the introductory Section 1.1.4. The description of the output is quite superficial there. Here it is explained more thoroughly.

The layout is common to many of the standard statistical tests, and a "dissection" is given in the following:

```
        One Sample t-test
```

This should be self-explanatory. It is simply a description of the test that we have asked for. Notice that, by looking at the format of the function call, `t.test` has automatically found out that a one-sample test is desired.

```
data:  daily.intake
```

This tells us which data are being tested. Of course, this will be obvious *unless* output has been separated from the command that generated it. This can happen, for example, when using the `source` function to read commands from an external file.

```
t = -2.8208, df = 10, p-value = 0.01814
```

This is where it begins to get interesting. We get the t statistic, the associated degrees of freedom, and the exact p-value. We do not need to use a table of the t distribution to look up which quantiles the t-value can be found between. You can immediately see that $p < 0.05$ and thus that (using the customary 5% level of significance) data deviate significantly from the hypothesis that the mean is 7725.

```
alternative hypothesis: true mean is not equal to 7725
```

This contains two important pieces of information: (a) the value we wanted to test whether the mean could be equal to (7725 kJ) and (b) that the test is two-sided ("not equal to").

```
95 percent confidence interval:
 5986.348 7520.925
```

This is a 95% confidence interval for the true mean; that is, the set of (hypothetical) mean values from which the data do not deviate significantly. It is based on inverting the t test by solving for the values of μ_0 that cause t to lie within its acceptance region. For a 95% confidence interval, the solution is

$$\bar{x} - t_{0.975}(f) \times \text{SEM} < \mu < \bar{x} + t_{0.975}(f) \times \text{SEM}$$

```
sample estimates:
mean of x
 6753.636
```

This final item is the observed mean; that is, the (point) estimate of the true mean.

The function `t.test` has a number of optional arguments, three of which are relevant in one-sample problems. We have already seen the use of `mu`

to specify the mean value μ under the null hypothesis (default is `mu=0`). In addition, you can specify that a one-sided test is desired against alternatives greater than μ by using `alternative="greater"` or alternatives less than μ using `alternative="less"`. The third item that can be specified is the *confidence level* used for the confidence intervals; you would write `conf.level=0.99` to get a 99% interval.

Actually, it is often allowable to abbreviate a longish argument specification; for instance, it is sufficient to write `alt="g"` to get the test against greater alternatives.

5.2 Wilcoxon signed-rank test

The t tests are fairly robust against departures from the normal distribution especially in larger samples, but sometimes you wish to avoid making that assumption. To this end, the *distribution-free methods* are convenient. These are generally obtained by replacing data with corresponding order statistics.

For the one-sample Wilcoxon test, the procedure is to subtract the theoretical μ_0 and rank the differences according to their numerical value, ignoring the sign, and then calculate the sum of the positive or negative ranks. The point is that, assuming only that the distribution is symmetric around μ_0, the test statistic corresponds to selecting each number from 1 to n with probability 1/2 and calculating the sum. The distribution of the test statistic can be calculated exactly, at least in principle. It becomes computationally excessive in large samples, but the distribution is then very well approximated by a normal distribution.

Practical application of the Wilcoxon signed-rank test is done almost exactly like the t test:

```
> wilcox.test(daily.intake, mu=7725)
        Wilcoxon signed rank test with continuity correction

data:  daily.intake
V = 8, p-value = 0.0293
alternative hypothesis: true location is not equal to 7725

Warning message:
In wilcox.test.default(daily.intake, mu = 7725) :
  cannot compute exact p-value with ties
```

There is not quite as much output as from `t.test` due to the fact that there is no such thing as a parameter estimate in a nonparametric test and therefore no confidence limits, etc., either. It is, however, possible under

some assumptions to define a location measure and calculate confidence intervals for it. See the help files for `wilcox.test` for details.

The relative merits of distribution-free (or *nonparametric*) versus parametric methods such as the t test are a contentious issue. If the model assumptions of the parametric test are fulfilled, then it will be somewhat more efficient, on the order of 5% in large samples, although the difference can be larger in small samples. Notice, for instance, that unless the sample size is 6 or above, the signed-rank test simply cannot become significant at the 5% level. This is probably not too important, though; what is more important is that the apparent lack of assumptions for these tests sometimes misleads people into using them for data where the observations are not independent or where a comparison is biased by an important covariate.

The Wilcoxon tests are susceptible to the problem of *ties*, where several observations share the same value. In such cases, you simply use the average of the tied ranks; for example, if there are four identical values corresponding to places 6 to 9, they will all be assigned the value 7.5. This is not a problem for the large-sample normal approximations, but the exact small-sample distributions become much more difficult to calculate and `wilcox.test` cannot do so.

The test statistic V is the sum of the positive ranks. In the example, the p-value is computed from the normal approximation because of the tie at 7515.

The function `wilcox.test` takes arguments `mu` and `alternative`, just like `t.test`. In addition, it has `correct`, which turns a continuity correction on or off (the default is "on", as seen from the output title; `correct=F` turns it off), and `exact`, which specifies whether exact tests should be calculated. Recall that "on/off" options such as these are specified using logical values that can be either `TRUE` or `FALSE`.

5.3 Two-sample t test

The two-sample t test is used to test the hypothesis that two samples may be assumed to come from distributions with the same mean.

The theory for the two-sample t test is not very different in principle from that of the one-sample test. Data are now from two groups, $x_{11}, \ldots, x_{1n_1}$ and $x_{21}, \ldots, x_{2n_2}$, which we assume are sampled from the normal distributions $N(\mu_1, \sigma_1^2)$ and $N(\mu_2, \sigma_2^2)$, and it is desired to test the null hypothesis $\mu_1 = \mu_2$. You then calculate

$$t = \frac{\bar{x}_2 - \bar{x}_1}{\text{SEDM}}$$

where the *standard error of difference of means* is

$$\text{SEDM} = \sqrt{\text{SEM}_1^2 + \text{SEM}_2^2}$$

There are two ways of calculating the SEDM depending on whether or not you assume that the two groups have the same variance. The "classical" approach is to assume that the variances are identical. With this approach, you first calculate a pooled s based on the standard deviations from the two groups and plug that value into the SEM. Under the null hypothesis, the t value will follow a t distribution with $n_1 + n_2 - 2$ degrees of freedom.

An alternative procedure due to Welch is to calculate the SEMs from the separate group standard deviations s_1 and s_2. With this procedure, t is actually not t-distributed, but its distribution may be approximated by a t distribution with a number of degrees of freedom that can be calculated from s_1, s_2, and the group sizes. This is generally not an integer.

The Welch procedure is generally considered the safer one. Usually, the two procedures give very similar results unless both the group sizes and the standard deviations are very different.

We return to the daily energy expenditure data (see Section 1.2.14) and consider the problem of comparing energy expenditures between lean and obese women.

```
> attach(energy)
> energy
   expend stature
1    9.21   obese
2    7.53    lean
3    7.48    lean
...
20   7.58    lean
21   9.19   obese
22   8.11    lean
```

Notice that the necessary information is contained in two parallel columns of a data frame. The factor `stature` contains the group and the numeric variable `expend` the energy expenditure in mega-Joules. R allows data in this format to be analyzed by `t.test` and `wilcox.test` using a model formula specification. An older format (still available) requires you to specify data from each group in a separate variable, but the newer format is much more convenient for data that are kept in data frames and is also more flexible if you later want to group the same response data according to other criteria.

The object is to see whether there is a shift in level between the two groups, so we apply a t test as follows:

```
> t.test(expend~stature)

        Welch Two Sample t-test

data:  expend by stature
t = -3.8555, df = 15.919, p-value = 0.001411
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -3.459167 -1.004081
sample estimates:
 mean in group lean mean in group obese
           8.066154           10.297778
```

Notice the use of the tilde (~) operator to specify that `expend` is *described by* `stature`.

The output is not much different from that of the one-sample test. The confidence interval is for the *difference* in means and does not contain 0, which is in accordance with the *p*-value indicating a significant difference at the 5% level.

It is Welch's variant of the *t* test that is calculated by default. This is the test where you do not assume that the variance is the same in the two groups, which (among other things) results in the fractional degrees of freedom.

To get the usual (textbook) *t* test, you must specify that you are willing to assume that the variances are the same. This is done via the optional argument `var.equal=T`; that is:

```
> t.test(expend~stature, var.equal=T)

        Two Sample t-test

data:  expend by stature
t = -3.9456, df = 20, p-value = 0.000799
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -3.411451 -1.051796
sample estimates:
 mean in group lean mean in group obese
           8.066154           10.297778
```

Notice that the degrees of freedom now has become a whole number, namely $13 + 9 - 2 = 20$. The *p*-value has dropped slightly (from 0.14% to 0.08%) and the confidence interval is a little narrower, but overall the changes are slight.

5.4 Comparison of variances

Even though it is possible in R to perform the two-sample t test without the assumption that the variances are the same, you may still be interested in testing that assumption, and R provides the `var.test` function for that purpose, implementing an F test on the ratio of the group variances. It is called the same way as `t.test`:

```
> var.test(expend~stature)

        F test to compare two variances

data:  expend by stature
F = 0.7844, num df = 12, denom df =  8, p-value = 0.6797
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.1867876 2.7547991
sample estimates:
ratio of variances
          0.784446
```

The test is not significant, so there is no evidence against the assumption that the variances are identical. However, the confidence interval is very wide. For small data sets such as this one, the assumption of constant variance is largely a matter of belief. It may also be noted that this test is not robust against departures from a normal distribution. The `stats` package contains several alternative tests for variance homogeneity, each with its own assumptions, benefits, and drawbacks, but we shall not discuss them at length.

Notice that the test is based on the assumption that the groups are independent. You should not apply this test to paired data.

5.5 Two-sample Wilcoxon test

You might prefer a nonparametric test if you doubt the normal distribution assumptions of the t test. The two-sample Wilcoxon test is based on replacing the data by their rank (without regard to grouping) and calculating the sum of the ranks in one group, thus reducing the problem to one of sampling n_1 values without replacement from the numbers 1 to $n_1 + n_2$.

This is done using `wilcox.test`, which behaves similarly to `t.test`:

```
> wilcox.test(expend~stature)

        Wilcoxon rank sum test with continuity correction

data:  expend by stature
W = 12, p-value = 0.002122
alternative hypothesis: true location shift is not equal to 0

Warning message:
In wilcox.test.default(x = c(7.53, 7.48, 8.08, 8.09, 10.15, 8.4,  :
  cannot compute exact p-value with ties
```

The test statistic W is the sum of ranks in the first group minus its theoretical minimum (i.e., it is zero if all the smallest values fall in the first group). Some textbooks use a statistic that is the sum of ranks in the *smallest* group with no minimum correction, which is of course equivalent. Notice that, as in the one-sample example, we are having problems with ties and rely on the approximate normal distribution of W.

5.6 The paired t test

Paired tests are used when there are two measurements on the same experimental unit. The theory is essentially based on taking differences and thus reducing the problem to that of a one-sample test. Notice, though, that it is implicitly assumed that such differences have a distribution that is independent of the level. A useful graphical check is to make a scatterplot of the pairs with the line of identity added or to plot the difference against the average of the pair (sometimes called a *Bland–Altman plot*). If there seems to be a tendency for the dispersion to change with the level, then it may be useful to transform the data; frequently the standard deviation is proportional to the level, in which case a logarithmic transformation is useful.

The data on pre- and postmenstrual energy intake in a group of women are considered several times in Chapter 1 (and you may notice that the first column is identical to `daily.intake`, which was used in Section 5.1). There data are entered from the command line, but they are also available as a data set in the `ISwR` package:

```
> attach(intake)
> intake
    pre post
1  5260 3910
2  5470 4220
3  5640 3885
4  6180 5160
```

```
5  6390 5645
6  6515 4680
7  6805 5265
8  7515 5975
9  7515 6790
10 8230 6900
11 8770 7335
```

The point is that the same 11 women are measured twice, so it makes sense to look at individual differences:

```
> post - pre
 [1] -1350 -1250 -1755 -1020  -745 -1835 -1540 -1540  -725 -1330
[11] -1435
```

It is immediately seen that they are all negative. All the women have a lower energy intake postmenstrually than premenstrually. The paired *t* test is obtained as follows:

```
> t.test(pre, post, paired=T)

        Paired t-test

data:  pre and post
t = 11.9414, df = 10, p-value = 3.059e-07
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 1074.072 1566.838
sample estimates:
mean of the differences
               1320.455
```

There is not much new to say about the output; it is virtually identical to that of a one-sample *t* test on the elementwise differences.

Notice that you have to specify `paired=T` explicitly in the call, indicating that you want a paired test. In the old-style interface for the unpaired *t* test, the two groups are specified as separate vectors and you would request that analysis by omitting `paired=T`. If data are actually paired, then it would be seriously inappropriate to analyze them without taking the pairing into account.

Even though it might be considered pedagogically dubious to show what you should *not* do, the following shows the results of an unpaired *t* test on the same data for comparison:

```
> t.test(pre, post) #WRONG!

         Welch Two Sample t-test

data:  pre and post
t = 2.6242, df = 19.92, p-value = 0.01629
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
  270.5633 2370.3458
sample estimates:
mean of x mean of y
 6753.636  5433.182
```

The number symbol (or "hash") `#` introduces a comment in R. The rest of the line is skipped.

It is seen that t has become considerably smaller, although still significant at the 5% level. The confidence interval has become almost four times wider than in the correct paired analysis. Both illustrate the loss of efficiency caused by not using the information that the "pre" and "post" measurements are from the same person. Alternatively, you could say that it demonstrates the gain in efficiency obtained by planning the experiment with two measurements on the same person, rather than having two independent groups of pre- and postmenstrual women.

5.7 The matched-pairs Wilcoxon test

The paired Wilcoxon test is the same as a one-sample Wilcoxon signed-rank test on the differences. The call is completely analogous to `t.test`:

```
> wilcox.test(pre, post, paired=T)
         Wilcoxon signed rank test with continuity correction

data:  pre and post
V = 66, p-value = 0.00384
alternative hypothesis: true location shift is not equal to 0

Warning message:
In wilcox.test.default(pre, post, paired = T) :
  cannot compute exact p-value with ties
```

The result does not show any material difference from that of the t test. The p-value is not quite so extreme, which is not too surprising since the Wilcoxon rank sum cannot get any larger than it does when all differences have the same sign, whereas the t statistic can become arbitrarily extreme.

Again, we have trouble with tied data invalidating the exact p calculations. This time it is the two identical differences of -1540.

In the present case it is actually very easy to calculate the exact p-value for the Wilcoxon test. It is the probability of 11 positive differences + the probability of 11 negative ones, $2 \times (1/2)^{11} = 1/1024 = 0.00098$, so the approximate p-value is almost four times too large.

5.8 Exercises

5.1 Do the values of the `react` data set (notice that this is a single vector, not a data frame) look reasonably normally distributed? Does the mean differ significantly from zero according to a t test?

5.2 In the data set `vitcap`, use a t test to compare the vital capacity for the two groups. Calculate a 99% confidence interval for the difference. The result of this comparison may be misleading. Why?

5.3 Perform the analyses of the `react` and `vitcap` data using nonparametric techniques.

5.4 Perform graphical checks of the assumptions for a paired t test in the `intake` data set.

5.5 The function `shapiro.test` computes a test of normality based on the degree of linearity of the Q–Q plot. Apply it to the `react` data. Does it help to remove the outliers?

5.6 The crossover trial in `ashina` can be analyzed for a drug effect in a simple way (how?) if you ignore a potential period effect. However, you can do better. Hint: Consider the intra-individual differences; if there were *only* a period effect present, how should the differences behave in the two groups? Compare the results of the simple method and the improved method.

5.7 Perform 10 one-sample t tests on simulated normally distributed data sets of 25 observations each. Repeat the experiment, but instead simulate samples from a different distribution; try the t distribution with 2 degrees of freedom and the exponential distribution (in the latter case, test for the mean being equal to 1). Can you find a way to automate this so that you can have a larger number of replications?