

10.9 Appendix: The Use of R for Two-Sample Comparisons

by EV Nordheim, MK Clayton & BS Yandell, November 4, 2003

This section shows a catalog of two-sample inference calculations using R. We begin with two-sample inference (testing and confidence intervals) under the usual assumptions. Then we show two-sample nonparametric tests, tests of proportions, and finally Levene's test for unequal variance.

10.9.1 Two-sample Inference based on the T-test

To perform a paired T -test with R, we could subtract the values for one treatment from the other and use the procedure we described in the appendix to Chapter 6. We illustrate using the data from the corn-breeding experiment from Section 10.2.

```
> backcross = c(209, 193, 223, 212, 238, 211, 228)
> inbred = c(202, 182, 221, 197, 233, 214, 218)
> t.test(backcross - inbred)
```

One Sample t-test

```
data: backcross - inbred
t = 2.951, df = 6, p-value = 0.02558
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 1.146891 12.281680
sample estimates:
mean of x
 6.714286
```

The `t.test` command gives the same answers we obtained by hand. (We note that this command gives us a confidence interval the difference.) Alternatively we can use both samples directly and the `paired` option to get the same result:

```
> t.test(backcross, inbred, paired = TRUE)
```

Paired t-test

```
data: backcross and inbred
t = 2.951, df = 6, p-value = 0.02558
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 1.146891 12.281680
```

```
sample estimates:
mean of the differences
      6.714286
```

The R command `t.test` can perform inference for two independent samples as well. In fact, the `t.test` command by default assumes you have independent samples (`paired=FALSE`). Note, however, that the `t.test` command will perform the test with the variances assumed unequal unless you specify otherwise. In order to perform the test with variance assumed equal, it is necessary to use the `var.equal=TRUE` option. To illustrate, we analyze the lizard data from Subsection 10.3.1 with Big Bend tail lengths in column `c4` and Box Canyon lengths in `c5`. The output for both the test and a 95% confidence interval is as follows.

```
> bigbend = c(6.2, 6.6, 7.1, 7.4, 7.6, 7.9, 8, 8.3, 8.4, 8.5, 8.6,
+ 8.8, 8.8, 9.1, 9.2, 9.4, 9.4, 9.7, 9.9, 10.2, 10.4, 10.8,
+ 11.3, 11.9)
> boxcanyon = c(8.1, 8.8, 9, 9.5, 9.5, 9.8, 9.9, 10.3, 10.4, 10.6,
+ 10.7, 10.9, 10.9, 11.1, 11.4, 12)
> t.test(bigbend, boxcanyon, var.equal = TRUE)
```

Two Sample t-test

```
data: bigbend and boxcanyon
t = -3.0933, df = 38, p-value = 0.003701
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -2.1266512 -0.4441821
sample estimates:
mean of x mean of y
 8.895833 10.181250
```

T-test using only sample mean and variance

Suppose you only had the sample sizes, means and the variances for two groups. You could compute the pooled variance, `sdpool`, as described in Section 10.3.1. Then you can compute the *t*-score and p-value in a similar manner to the one-sample test shown in Appendix 6.

The `t.test` command does not actually show you the pooled standard deviation. You can compute the pooled variance as follows given the sample sizes and variances:

```
> n1 = length(bigbend)
> n2 = length(boxcanyon)
> var1 = var(bigbend)
> var1
```

```
[1] 2.044764
```

```

> var2 = var(boxcanyon)
> var2

[1] 1.064292

> sdpool = sqrt(((n1 - 1) * var1 + (n2 - 1) * var2)/(n1 + n2 -
+ 2))
> sdpool

[1] 1.287531

```

The t -value can be computed from the sample means, pooled SD, and samples sizes as follows:

```

> ybar1 = mean(bigbend)
> ybar1

[1] 8.895833

> ybar2 = mean(boxcanyon)
> ybar2

[1] 10.18125

> t.value = (ybar1 - ybar2)/(sdpool * sqrt((1/n1) + (1/n2)))
> t.value

[1] -3.093299

```

Again, as in Appendix 6, we find the p-value using the `pt` command:

```

> 2 * pt(abs(t.value), n1 + n2 - 2, lower.tail = F)

[1] 0.003701436

```

Note that this agrees with the calculation above using the `t.test`.

T-test with unequal variances

To perform independent two-sample inference with the variances assumed unequal, we omit the `var.equal` option (the default is `FALSE`). For example, consider the compost data from Subsection 10.3.2 on the percent germination for composts A and B. The resultant commands and output are as follows.

```

> compostA = c(24, 25, 26, 26, 27, 28, 28, 30, 33)
> compostB = c(22, 32, 37, 40, 44, 47, 49, 51, 52, 56, 67)
> t.test(compostA, compostB)

```

Welch Two Sample t-test

```
data:  compostA and compostB
t = -4.6659, df = 11.214, p-value = 0.0006528
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -26.084991  -9.389757
sample estimates:
mean of x mean of y
 27.44444  45.18182
```

10.9.2 Two-sample Nonparametric Test

The Mann-Whitney, or Wilcoxon, test is available in R as `wilcox.test` when you are not willing to assume normality. It works in a similar fashion to the `t.test`, which we illustrate with the DMBA drug study of Section 10.5.1.

```
> drugA = c(14, 11, 37, 17, 21)
> drugB = c(8, 13, 9)
> wilcox.test(drugA, drugB)
```

Wilcoxon rank sum test

```
data:  drugA and drugB
W = 14, p-value = 0.07143
alternative hypothesis: true mu is not equal to 0
```

The Wilcoxon "statistic" $W=14$ is slightly different from the Mann-Whitney statistic. It is the sum of the ranks for drugA (29) minus the smallest possible sum of ranks ($15 = 1+2+3+4+5$). This command can also be used for paired samples, such as the corn breeding data. It can either be used on the difference or on the pairs:

```
> wilcox.test(backcross - inbred)
```

Wilcoxon signed rank test

```
data:  backcross - inbred
V = 26, p-value = 0.04688
alternative hypothesis: true mu is not equal to 0
> wilcox.test(backcross, inbred, paired = TRUE)
```

Wilcoxon signed rank test

```
data:  backcross and inbred
V = 26, p-value = 0.04688
alternative hypothesis: true mu is not equal to 0
```

10.9.3 Test of Two Proportions

Two proportions can be compared with the R command `prop.test` already seen for one sample in Appendices for Chapters 6 and 7. For the fungicide experiment of Section 10.7.1, we can do

```
> prop.test(c(54, 40), c(72, 67))

      2-sample test for equality of proportions with continuity correction

data:  c(54, 40) out of c(72, 67)
X-squared = 3.0442, df = 1, p-value = 0.08103
alternative hypothesis: two.sided
95 percent confidence interval:
 -0.01568796  0.32165810
sample estimates:
   prop 1    prop 2 
0.7500000 0.5970149
```

By default, this uses the continuity correction. We can avoid this using the `correct=FALSE` option, yielding a p-value of 0.054, which agrees with the value found in Section 10.7.1.

10.9.4 Levene's two-sample test of unequal variance

Unfortunately, R does not currently have a single command for performing Levene's test for unequal variance. As a result at least part of these computations of Section 10.4 must be done "by hand." For the adventurous, here is a home-made command in R to do all the calculations for Levene's test:

```
> levene.test = function(data1, data2) {
+   levene.trans = function(data) {
+     ## find median for group of data
+     ## subtract median; take absolute value
+     a = sort(abs(data - median(data)))
+     ## if odd sample size, remove exactly one zero
+     if (length(a)%%2)
+       a[a != 0 | duplicated(a)]
+     else a
+   }
+   ## perform two-independent sample T-test on transformed data
+   t.test(levene.trans(data1), levene.trans(data2), var.equal = TRUE)
+ }
```

The calculations in this new command `levene.test` are compact but subtle, not for everyone. However, for those interested, here is an explanation. Inside the `levene.test` command definition is another definition, for `levene.trans`. This does steps 1-3 of Levene's test. First, we take the absolute value of the difference between the data and its median. Then, if the sample size n is odd, we keep all data except the first zero. (The `n%2` "command" reads " n modulo 2", which is 0 if n is even and 1 if n is odd.)

We illustrate Levene's test with the bean seed planter data of Section 10.4:

```
> planterI = c(1.2, 1.6, 1.7, 2.4, 2.4, 2.5, 2.6, 2.7, 2.7, 2.9,
+ 3.1, 3.2, 3.3, 3.7, 3.9, 4)
> planterII = c(2.2, 2.3, 2.3, 2.5, 2.5, 2.5, 2.6, 2.7, 2.7, 2.8,
+ 2.8, 2.8, 2.8, 2.9, 3, 3.1, 3.1, 3.3, 3.4)
> levne.test(planterI, planterII)
```

Two Sample t-test

```
data: levne.trans(data1) and levne.trans(data2)
t = 2.6083, df = 32, p-value = 0.01372
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.07316947 0.59488609
sample estimates:
mean of x mean of y
0.6062500 0.2722222
```

The original calculations had $0.01 \leq \text{p-value} \leq 0.02$, while we now have $\text{p-value} = 0.014$.