Alternatives to the t-test

Mandy Vogel

July 18, 2015

Table of Contents

Permutation Tests

- does not rely on a assumed a priori distribution
- instead a empirical distribution is created from randomization of observed data
- robust against deviations from normality
- sensitive to differences in treatment variances
- scope of inference is limited to the sample (importance has been questioned by a number of authors)

Excurse - tapply()

- `tapply()` allows you to create tables (hence the "t") of the value of a function on subgroups defined by its second argument, which can be a factor or a list of factors.
- e.g. in the `sleep` data frame, we can summarize `extra` (increase in hours of sleep) classify by `group` (drug given) as follows:

Input/Output

```
> tapply(X = sleep$extra, INDEX = sleep$group, FUN = mean)
   1    2
0.75 2.33
> tapply(sleep$extra,sleep$group,mean)
   1    2
0.75 2.33
```

- `lapply()` is a member of the apply functions

Excurse - `replicate()`

- the function `replicate()` is also a member of the apply family
- it is used for repeated evaluation of an expression (usually in the context of random number generation)
- example: roll 6 dice and sum them up 10000 times

Excurse - replicate()

Input/Output

```
> sample(1:6,size = 6, replace = T)
[1] 2 6 2 3 5 3
> sum(sample(1:6,size = 6, replace = T))
[1] 19 ## not the same sample as above
> dice <- replicate(10000,sum(sample(1:6,size = 6,replace = T)))
> table(dice)
dice
  6   7   8   9  10  11  12  13  14  15  16  17  18  19 ...
  1   1   3   7  16  49  80 163 274 357 494 602 723 842 ...
 26  27  28  29  30  31  32  33  34
483 356 270 170 108  63  33  11   4
```

Permutation Tests - Example

- we use the `sleep` data set
- first we calculate the difference of the group means using the functions `lapply()` and `diff()`
- we use the absolute difference (this corresponds to a two-sided test)

Permutation Tests - Example

Input/Output

```
## use tapply to calculate the group means (not necessary)
> tapply(sleep$extra,sleep$group,FUN = mean)
   1    2
0.75 2.33
> ## calculate the difference between the group means
> ## (not necessary)
> diff(tapply(sleep$extra,sleep$group,FUN = mean))
   2
1.58
> ## calculate the absolute difference (two-sided test)
> orig.diff <- abs(diff(tapply(sleep$extra,sleep$group,mean)))
> orig.diff
   2
1.58
```

Permutation Tests - Example

Input/Output

```
> ## do it a 10000 times
> res <- replicate(10000,
+    abs(diff(tapply(sleep$extra,sample(sleep$group),mean))))
> sum(orig.diff <= res)/10000 ## p-value
[1] 0.0724
```

Permutation Tests - Example

Input/Output

```
> ## compare with t-test
> t.test(sleep$extra ~ sleep$group)

Welch Two Sample t-test

data:  sleep$extra by sleep$group
t = -1.8608, df = 17.776, p-value = 0.07939
alternative hypothesis: true difference in means is not equal to
95 percent confidence interval:
 -3.3654832  0.2054832
sample estimates:
mean in group 1 mean in group 2
           0.75            2.33
```

Rank-Based Permutation Tests

- requiring a single null distribution given a particular sample size
- much less sensitive to outliers compared to parametric methods
- scope of inference is generally less considered an issue
- slightly less powerful than parametric methods if their parametric assumptions hold
- computational problems of ties
- rank transformation throws out a large amount of information
- also sensitive to heteroscedasticity

Rank-Based Permutation Tests

Warnings

- these tests are not free from assumptions
- hypotheses statements will generally differ from parametric tests

Rank-Based Permutation Tests - Example

- consider to hypothesized populations X_1 and X_2
- assume two observations for X_1 and three observations for X_2
- one-tailed test: $\mathrm{H}_0 : \mathrm{X}_1 \geq X_2$ versus $\mathrm{H}_A : \mathrm{X}_1 < X_2$
- in absence of ties we have $(n_1 + n_2)!/(n_1!n_2!) = 10$ possiple ranks for X_1 or X_2
- what is the smallest possible p-value?

Wilcoxon tests

- the Wilcoxon tests the location of the median
- it is a non-parametric alternative to Student's t test
- it is based on the ranks – and really simple, e.g. for the two sample test: sort your data, give them ranks, sum up the ranks by group, take the smaller sum and look in a table for the appropriate row/column (with ties are dealt with by averaging the appropriate ranks)
- in R it is (not very surprisingly) `wilcox.test()`
- there is also a one and a two sample and a paired version

Wilcoxon Test

- the non-parametric test is much more appropriate when the errors are not normal,
- can be more powerful if the distribution is strongly skewed by the presence of outliers
- typically the t-test will give the lower p-value, so the Wilcoxon test is said to be conservative: if a difference is significant under a Wilcoxon test it would have been even more significant under a t-test.

Wilcoxon Signed Rank Test

- method for one sample or two dependent samples
- calculate the differences between the pairs of observations (or between values and hypothesized value)
- let n be the number of non-zero differences
- rank the absolute values of the n differences
- reassign the signs from step 1
- T_+ is the sum of the positive signed ranks, while T_- is the sum of the negative ranks
- for a two tailed test the minimum of these two is taken as test statistic, in a upper-tailed T_-, in a lower-tailed T_+

Wilcoxon Signed-Rank Test

```
> pre.test <- c(17,12,20,12,20,21,23,10,15,17,18,18)
> post.test <- c(19,25,18,18,26,19,27,14,20,22,16,18)
> wilcox.test(pre.test,post.test,paired = T)

Wilcoxon signed rank test with continuity correction

data:  pre.test and post.test
V = 7.5, p-value = 0.02527
alternative hypothesis: true location shift is not equal to 0

Warning messages:
1: In wilcox.test.default(pre.test, post.test, paired = T) :
  kann bei Bindungen keinen exakten p-Wert Berechnen
2: In wilcox.test.default(pre.test, post.test, paired = T) :
  kann den exakten p-Wert bei Nullen nicht berechnen
```

Exercise

1. load the data frame normtemp from the file temperature.rdata; it contains the body temperature of several individuals, the gender and the heart rate
2. test if the temperature is different in male (coded as 1) and female (coded as 2), use the appropriate test.
3. test again, compare the results of the t test and the wilcoxon.
4. plot the respective boxplots!