Displays for Statistics 5303

Lecture 4

September 11, 2002

Christopher Bingham, Instructor

612-625-7023 (St. Paul)
612-625-1024 (Minneapolis)

Class Web Page
http://www.stat.umn.edu/~kb/classes/5303

Randomization form of 2 sample t

When you have two independent random samples, one of size n_1 from $N(\mu_1, \sigma)$ and the other of size n_2 from $N(\mu_1, \sigma)$, one standard way to test H_0: $\mu_1 = \mu_2$ (treatments have same effect) uses the **two sample t-statistic**

$$t = (\overline{y_2}_\bullet - \overline{y_1}_\bullet)/\hat{SE}[\overline{y_2}_\bullet - \overline{y_1}_\bullet],$$

where

$$\hat{SE}[\overline{y_2}_\bullet - \overline{y_1}_\bullet] = s_p\sqrt{\{1/n_1 + 1/n_2\}}$$

$$s_p^2 = \{(n_1-1)s_1^2 + (n_2-1)s_2^2\}/(n_1 + n_2 - 2)$$

$$= \{\sum(y_{1i}-\overline{y_1}_\bullet)^2+\sum(y_{2i}-\overline{y_2}_\bullet)^2\}/(n_1 + n_2 - 2)$$

Under H_0, t is Student's t on n_1+n_2-2 d.f.

s_p^2 is a "pooled" estimate of σ^2.

This form of the two sample t is also known as (R. A.) Fishers' t statistic

Because of the assumption of equal variances, H_0 implies the two distributions are the same, a stronger assertion than that the means are the same.

The two-sample t is also standard in a randomized experiment with N EU's, n_1 of which are randomly assigned to treatment 1 and the remainder $n_2 = N - n_1$ are assigned to treatment 2. Unless the treatments differentially affect variability, in a randomized experiment, the equality of variances is usually plausible.

Inference based on the randomization does not require the assumption of normality.

Here is an analysis of an experiment comparing the coefficients of digestibility (in percent) of dry matter feed corn silage fed to 7 sheep and 6 steers. The interest was whether there was a difference between the groups.

In honesty, this is not a good example in which to use randomization inference, because it *could* not be randomized. At best the two sample can be considered random samples from populations of sheep and steers.

```
Cmd> sheep <- enter(578 562 619 544 536 564 532)/10

Cmd> steers <- enter(642 587 631 625 598 592)/10

Cmd> stuff <- t2val(sheep,steers,df:T); stuff
component: t
(1)     -3.3442
component: df
(1)         11

Cmd> twotailt(stuff$t, stuff$df) # 2-tail P-value
(1)   0.0065449
```

$$H_0: \mu_{sheep} = \mu_{steers}, H_a: \mu_{sheep} \neq \mu_{steers}$$

The P-value < .01 indicates a significant difference of means.

As for the paired situation, the randomization test is conditional on the observed set of $N = n_1 + n_2$ values

$$y_{11}, \ldots, y_{1n_1}, y_{21}, \ldots, y_{2n_2}$$

If the two treatment groups are truly equivalent, and treatments have no differential effect (in *mean*, *variability* or *anything else*), then the subscript labelling the group is irrelevant.

Since this labelling was selected randomly, the test compares the observed value of t with the set of possible values that would have been obtained for different randomizations.

In principle, you can compute t for each possible split into samples. If the actual t is extreme enough relative to the set of all values found, you reject the hypothesis that the treatments had the same effect.

With the paired t-statistic it was easier to work with the simpler $\sum \pm d_i$. Similarly, with the 2 sample test it is easier to work with $\overline{y_{2\bullet}} - \overline{y_{1\bullet}}$.

Algebra shows that t can be expressed as

$$t = \sqrt{(n_1 + n_2 - 2)}\, r / \sqrt{\{1 - r^2\}}$$

where

$$r = \sqrt{\{n_1 n_2/(n_1+n_2)\}}\, (\overline{y_{2\bullet}} - \overline{y_{1\bullet}})/\sqrt{SS}$$

with

$$SS = \sum (y_{i1} - \overline{y_{\bullet\bullet}})^2 + \sum (y_{i2} - \overline{y_{\bullet\bullet}})^2$$

Since $\sqrt{(n_1 + n_2 - 2)}$ and $\sqrt{\{n_1 n_2/(n_1+n_2)\}}$ are constants and SS is the same for any split into groups, conditional on the data, t is a function of $\overline{y_{2\bullet}} - \overline{y_{1\bullet}}$.

So to do the randomization test you compute $\overline{y_{2\bullet}} - \overline{y_{1\bullet}}$ for all possible splits of the data and compare the observed value of $\overline{y_{2\bullet}} - \overline{y_{1\bullet}}$ with it.

Now there are

$$\binom{n_1 + n_2}{n_1} = (n_1+n_2)!/(n_1!n_2!)$$

possible samples. This *binomial coefficient* can be huge.

```
Cmd> n1 <- nrows(sheep); n2 <- nrows(steers); N <- n1 + n2

Cmd> binom(N,n1) # number of possible assignments
WARNING: searching for unrecognized macro binom near binom(
(1)       1716   13!/(7!6!)
```

This is too many to do by hand. And the code for generating the different samples is a little tricky, so I immediately use MacAnova macro **randt2()**.


```
Cmd> actual <- describe(sheep,mean:T)-describe(steers,mean:T)

Cmd> actual # observed ybar_sheep - ybar_steers
(1)     -5.0357

Cmd> usage(randt2)
randt2(y1, y2 [,trials:n]), REAL vectors y1 and y2, positive
integer n

Cmd> diffs <- randt2(sheep,steers) 3 all possible values
WARNING: searching for unrecognized macro randt2 near
stuff <- randt2(

Cmd> M <- length(diffs); M # it has the right length
(1)        1716

Cmd> sum(diffs == actual) # observed obtained just once
(1)           1

Cmd> sum(diffs <= actual) # 9 out of 1716 are <= observed
(1)           9

Cmd> sum(stuff <= actual)/M # one tail P-value
(1)   0.0052448

Cmd> 2*sum(stuff <= actual)/M # 2 tail P-value 18/1716
(1)     0.01049
```

The normal theory P-value was .0065449, very much in the same ball park.

When $(n_1+n_2)!/n_1!n_2!)$ is impractically large, you can estimate the P-value by randomly sampling the set of $(n_1+n_2)!/(n_1!n_2!)$ possible assignments to treatments using keyword **trials**.

```
Cmd> M <- 5000 # number of random assignments
Cmd> diffs <- randt2(sheep,steers,trials:M)
Cmd> p <- sum(diffs <= actual)/M; p
(1)      0.0038
Cmd> 2*p
(1)      0.0076
```

The number $\leq$ the actual $\overline{y}_{2\bullet} - \overline{y}_{1\bullet}$ has a binomial distribution so the estimated pvalue p has standard error $\sqrt{\{p(1-p)/M\}}$

```
Cmd> 1.96*sqrt(p*(1 - p)/M) # margin of error for p
(1)   0.0017054
```


To reduce the margin of error in estimating the p-value, you can increase the size of the simulation.

```
Cmd> M <- 20000; diffs <- randt2(sheep,steers,trials:M)
Cmd> p <- sum(diffs <= actual)/M; p
(1)     0.00545
Cmd> 2*p
(1)      0.0109
```

This is quite close to the exact randomization P-value 0.01049.

Now the margin of error is.

```
Cmd> 1.96*sqrt(p*(1 - p)/M)
(1)   0.0010204
```

Completely Randomized Design (CRD)

This is the simplest design for comparing treatments and is sometimes the preferred design.

Setup You have g treatments to compare and N EU's available. By some process that we will discuss at another time, you have determined you want sample sizes $n_1, n_2, \ldots, n_g$ for the treatments, with $\sum n_i = N$.

CRD assignment

1 Randomly choose n_1 EU's for treatment 1 (say first n_1 numbers to be drawn from a box with 1, 2, ..., N).

2 Randomly choose n_2 EU's from the remaining for treatment 2,

....

g-1 Randomly choose n_{g-1} EU's for treatment g-1

g Remaining n_g EU's are assigned to treatment g

This amounts to selecting a random permutation (reordering) of

$$\{1, 2, \ldots, N = n_1 + n_2 + \ldots + n_g\},$$

just as in the two sample case.

Advantages of CRD:

- Simple to do; all you need is one random permutation of {1, 2, ..., N}
- Relatively simple to analyze
- No restrictions on n_i's. Some other designs severely restrict n_i's.
- If you end up with missing values, *where the reason for them being missing is unrelated to treatment*, the analysis is still easy

Disadvantages of CRD:

- The accuracy of the experiment depends on the variability among all the experimental units, not on the variability among subsets.
- Because of the greater variability you may require more EU's than with other more complicated designs.

To test the hypothesis that there were no treatment effects, the standard analysis is a **one-way ANOVA** (Analysis of Variance) .

If the treatment levels are **qualitative** (qualitative), you almost always follow up the ANOVA with a **multiple comparisons analysis** to try to pinpoint exactly which treatments differed from which.

If the treatment levels are **quantitative** (for example a dosage, a temperature, dollars spent per child), your follow up may include fitting a model such as a linear (straight line) dependence of means on the variable defining the treatments or possible some curved dependence, perhaps quadratic.

Randomization inference is also applicable in this situation.

There are $N!/(n_1!n_2!...n_g!)$ possible assignments of the EU's to the groups. In principle, you can compute an F-statistic for each such assignment. Then you compare the actual F-statistic with this set of possible F's. If it is extreme enough you reject the null hypothesis that the treatments had the same effect.

The null hypothesis is that the treatments have no influence on the observed values so the labelling in groups is irrelevant.

By similar algebra as for two-sample t, you can show that a randomization test using F is equivalent to one which rejects H_0 for large $SS_{trt} = \sum n_i(\overline{y_{i\bullet}} - \overline{y_{\bullet\bullet}})^2$.

If the treatments really have the same effect,all the groups should be about equally variable, just as for the normal theory test which assumes $\sigma_1 = \sigma_2 = ... = \sigma_g$.

Also, the closer the observed distributions of response are to normal, the closer the easily computed normal theory P-values are to randomization-based P-values which often can only be estimated by simulation.

For both these reasons, some preliminary examination of your data is almost always called for to check assumptions. If they don't hold, sometimes you can work with some transformation of the data such as log(y) or $\sqrt{y}$.

Example 3.2:

This analysis data consisting of the log times until failure of a resin under stress in accelerated life tests. There were 5 treatments determined by temperature. See Table 3.1.

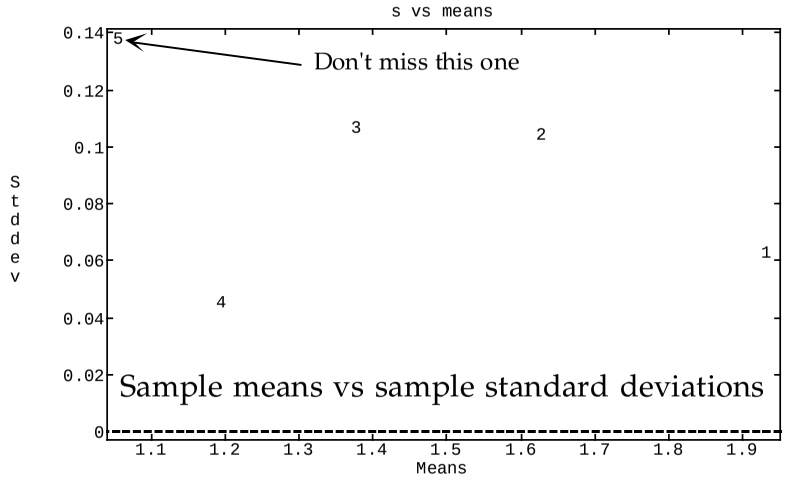
Here I read the data from file `OeCh03.dat` containing Chapter 3 data sets in a form readable by `read()` or `matread()`. First I read information on data sets in the file:

```
Cmd> read("","info") # or matread("","info")
info  0
) Data sets for Chapter 3 of Oehlert's A First Course in Design
) and Analysis of Experiments examples and exercises.
)
) Data set names for examples, exercises, and problems have the
) form exmplC.N, exC.N, or prC.N where C is the chapter number
) and N is the example/exercise/problem number.  For example
) ex20.2 is Exercise 2 inChapter 30.
)
) The names of data sets in the file are
) exmpl3.2 (resin lifetimes)
) ex3.1 (rat liver weights)
) ex3.3 (orange pulp silage)
) ex3.5 (leaf angles)
) pr3.1 (solder joints)
) pr3.2 (fruit fly longevity)
) pr3.3 (alpine meadows)
) pr3.4 (caffeine/adenine)
) pr3.5 (polypropylene fibers)
WARNING: 0 lines of data in data set
Read from file "TP1:Stat5303:Data:OeCh03.dat"
```

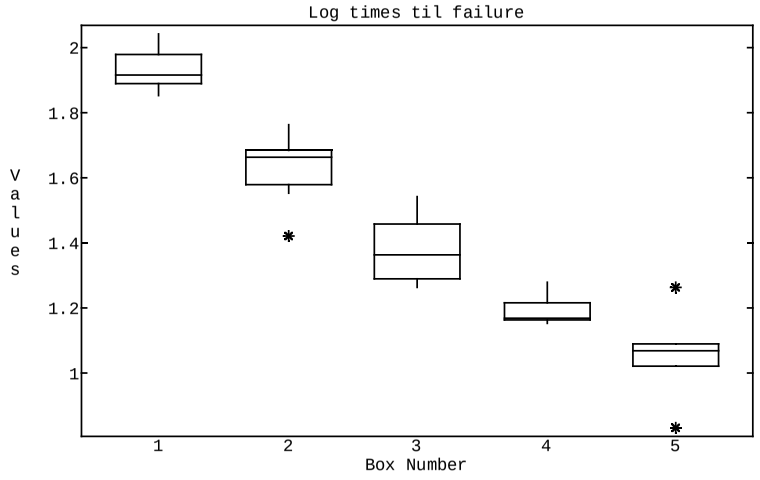

This shows the data set name is `exmpl3.2`

```
Cmd> data <- read("","exmpl3.2")
exmpl3.2          37     2
) A data set from Oehlert (2000) \emph{A First Course in Design
) and Analysis of Experiments}, New York: W. H. Freeman.
)
) Data originally from Kvam, P. H. and Samaniego, F. J. (1993).
) ``Life Testing in Variably Scaled Environments.'' {\em
Technometrics} 35, 306--314.
)
) Table 3.1, p. 33
) These are the log10 times to failure (in hours) of a resin
under five
) different temperature stresses.  Column 1 is) temperature
(levels 1
) through 5 are 175, 194, 213, 231, 250) degrees C, and Column 2
is response.
Read from file "TP1:Stat5303:Data:OeCh03.dat"

Cmd> data[run(10),] # first 10 cases
 (1,1)           1        2.04
 (2,1)           1        1.91
 (3,1)           1           2
 (4,1)           1        1.92
 (5,1)           1        1.85
 (6,1)           1        1.96
 (7,1)           1        1.88
 (8,1)           1         1.9
 (9,1)           2        1.66
(10,1)           2        1.71

Cmd> treat <- factor(data[,1]) # make column 1 a factor

Cmd> logy <- data[,2] # response is column 2

Cmd> stats <- tabs(logy,treat,mean:T,count:T,stddev:T)

Cmd> stats # some statistics for each group
component: mean   Treatment sample means
(1)      1.9325      1.6287      1.3775      1.1943      1.0567
component: count  Treatment sample sizes
(1)           8           8           8           7           6
component: stddev Treatment sample standard deviations
(1)    0.063415      0.1048     0.10714    0.045774     0.13837
```

```
Cmd> plot(stats$mean,stats$stddev,\
     title:"s vs means",symbols:run(5),\
     xlab:"Means",ylab:"Stddev",ymin:0)
```




```
Cmd> vboxplot(split(logy,treat),title:"Log times til failure")
```



There isn't much visual evidence of non constant standard deviation.

We can see what sort of dependence the means have on temperature.

```
Cmd> temperature <- vector(175, 194, 213, 231, 250)
```

```
Cmd> plot(temperature,stats$mean,\
     title:"Mean log lifetimes vs temperature",\
     xlab:"Temperature",ylab:"Means")
```