Displays for Statistics 5303

Lecture 5

September 13, 2002

Christopher Bingham, Instructor

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

Class Web Page

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

Reprise of Example 3.2:

Data on 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.

I first read the info record on file **OeCh03.dat** to find the data set name and then and then read the data set and split it into a factor and a response.

```
Cmd> read("","info") # read from file of named data sets
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"
```



```
Cmd> data <- read("","exmpl3.2")# data has 37 cases, 2 vars
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]) # 1st variable is treatment

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

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

Cmd> stats
component: mean
(1)      1.9325      1.6287      1.3775      1.1943      1.0567
component: count
(1)           8           8           8           7           6
component: stddev
(1)    0.063415      0.1048     0.10714    0.045774     0.13837
```


All formal statistical analysis is based on probability models, usually best described in the language of mathematics.

In designed experiments, the model usually consists of two parts.

- A part describing the means
- A part describing the "errors", that is, deviations of responses from means

The usual one-way ANOVA model with no special restrictions on the means is:

- There are unknown means $\mu_1, \ldots, \mu_g$ for each group (model for means)
- Errors $\varepsilon_{ij} = y_{ij} - \mu_i$ are *independent* normal $N(0,\sigma^2)$ (model for errors)

An observation $y_{ij} = \mu_i + \varepsilon_{ij}$.
This is a particular case of an *additive* decomposition of a response into a *predictable* part (μ_i) and an *unpredictable part* (ε_{ij}).

Alternatively, you can summarize both parts of the model simultaneously by

$$y_{ij} \text{ are independent } N(\mu_i, \sigma^2)$$

The most important feature of this model is that all errors are independent of each other.

The next most important feature of this model is that the standard deviation does not depend on the treatment, that is, σ is constant

Another feature, usually less important, is that the errors are normal.

Some features of the model such as constant σ and normality are checkable to some extent.

Others such as independence are very difficult or impossible to check, but are effectively guaranteed by proper randomization. This is another reason randomization is important.

Why do we care about models and whether the data is consistent with a model?

Because statistical procedures are developed to "work" in an environment in which certain assumptions are true. And many procedures do not "work" in situations where these assumptions are false.

What does it mean for a statistical procedure to work?

Significance test

The *actual* significance level of a significance or hypothesis test is defined as

$$\alpha = P(\text{reject } H_0 \mid H_0 \text{ true})$$

When you do a significance or hypothesis test you always have an *intended* significance level say .05 or .01. If the actual significance level $\neq$ intended, the significance tests *is not working*.
Example: You choose .05 and P(reject $H_0 \mid H_0$ true) = .11 $\neq$ .05 $\Rightarrow$ *not working*.

Confidence intervals

A defining property of a 95% confidence interval procedure, say, is that

$$P(\text{interval surrounds parameter}) \cong .95$$

When you calculate a C.I., you always have an *intended* confidence level, often 95% or 99%.

A C.I. *doesn't work* when the *actual confidence level* =
P(interval surrounds parameter) $\neq$
intended confidence level.

Example Suppose you do the calculation for a 95% interval (say $\mu = \overline{y} \pm 2.228 \times s/\sqrt{n}$ for a mean based on n = 11 observations).

If the actual confidence level = 89.5% or 99.1%, the confidence interval *isn't working*.

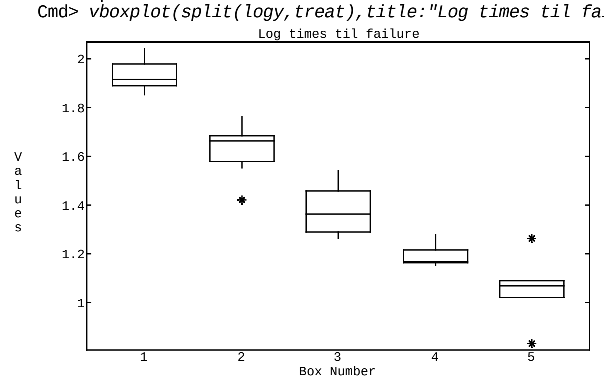
Returning to the ANOVA model:

An important part of the model is that the standard deviations are the same in each group:

$$\sigma_1 = \sigma_2 = \ldots = \sigma_g$$

You can get some information about this from plots.

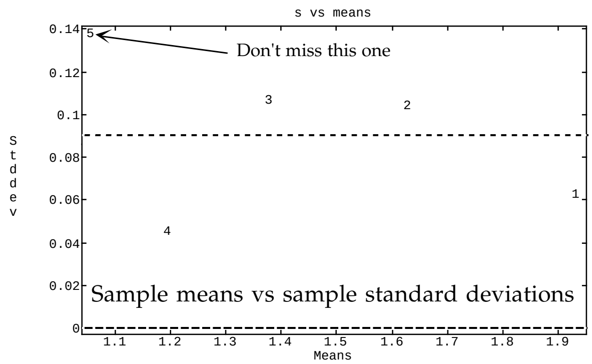
```
Cmd> vboxplot(split(logy,treat),title:"Log times til failure")
```



There is no obvious pattern. The highest temperature group may have a couple of outliers, but it's hard to say, because the groups size is small.

Another useful plot is of standard deviations s_i vs means $\overline{y}_i$:

```
Cmd> stats <- tabs(logy,treat,mean:T,count:T,stddev:T)
Cmd> plot(stats$mean,stats$stddev,\
     title:"s vs means",symbols:run(5),\
     xlab:"Means",ylab:"Stddev",ymin:0)
```



The dashed line is rather arbitrary, but probably describes the pattern as well as any other line or curve. This is what you hope to see - scatter of the points around a horizontal line.

It's hard to formally check the assumption of equal σ. Here a way using simulation, based on

$$\max(s_i)/\min(s_i) = \sqrt{F_{max}}$$

where F_{max} $\max(s_i^2)/\min(s_i^2)$ is Hartley's maximum F statistic:

```
Cmd> N <- sum(stats$count);N
(1)       37
Cmd> max(stats$stddev)/min(stats$stddev) # observed
(1)      3.0229
```

Is this unusually large? We can find out by simulation. I first do 5000 repetitions with normal data

```
Cmd> M <- 5000; ratio <- rep(0,M) # place to put results
Cmd> for(i,1,M){
     @temp <- tabs(rnorm(N),treat,stddev:T)
     ratio[i] <- max(@temp)/min(@temp);;}
Cmd> sum(ratio >= 3.0229)/M # P-value
(1)      0.0842   proportion ≥ observed
```

This test depends strongly on normality. Here I do 5000 repetitions with t_{10} data

```
Cmd> for(i,1,M){
     @temp <- tabs(invstu(runi(N),10),treat,stddev:T)
     ratio[i] <- max(@temp)/min(@temp);;}
Cmd> sum(ratio >= 3.0229)/M
(1)      0.1444
```

The P-value is quite different.

Means are often not of interest

When your interest is in comparing treatments, the means μ_i themselves are usually not of great interest, since they often depend on the specific details of the experiment such as time of year, location, even time of day.

What you should be interested in is the **effect** of one treatment as compared to another. Because of this, almost always the μ_i's and are expressed in another way:

$$\mu_i = \mu^* + \alpha_i$$

where μ^* is a number summarizing the overall level of the response regardless of treatment and

$$\alpha_i = \mu_i - \mu^*$$

is the "effect" of the treatment, the amount the mean μ_i is changed by the treatment from the overall level μ^*.

A confusing issue is that there is no single way to do this, because the actual values of the α_i's depend on what value you take for μ^*.

For many purposes, the definition of μ^* doesn't really matter, because

$$\mu_i - \mu_j = \mu^* + \alpha_i - (\mu^* + \alpha_j) = \alpha_i - \alpha_j$$

no matter what μ^* is.

And more generally, if w_1 w_2, ..., w_g are a set of numbers that define a *contrast* among the means, that is $\sum w_i = 0$,

$$\sum w_i\mu_i = (\sum w_i)\mu^* + \sum w_i\alpha_i = \sum w_i\alpha_i$$

which doesn't depend on μ^*, reducing to a contrast among the effects α_i.

There are several fairly standard choices for μ^*:

1 $\mu^* = \sum\mu_i/g$ = unweighted average of μ_i

For this choice, $\sum\alpha_i = 0$.

This is used by MacAnova in its computations and is associated with the name Scheffe.

2 $\mu^* = \sum n_i\mu_i/N$ = weighted average of μ_i, where $N = \sum n_i$

For this choice $\sum n_i\alpha_i = 0$.

In cases when the n_i's are somewhat accidental, this probably doesn't make much sense, but it has some mathematical advantages.

3 $\mu^* = \mu_j$ with $\alpha_j = 0$, $\alpha_i = \mu_i - \mu_j$, $i \neq j$

Particular cases are

$\mu^* = \mu_1$, with $\alpha_1 = 0$, $\alpha_i = \mu_i - \mu_1$

$\mu^* = \mu_g$, with $\alpha_g = 0$, $\alpha_i = \mu_i - \mu_g$

Each treatment is compared with a particular treatment.

This approach is particularly appropriate when the specific treatment others are compared to is a **control**.

I believe SAS uses $\mu^* = \mu_g$ and GLIM uses $\mu^* = \mu_1$ in computation.

In using a computer program which purports to compute "effects", it's important to know their definition.

Estimates of means and effects are fairly obvious.

Each μ_i is estimated by a sample mean

$$\hat{\mu}_i = \overline{y}_{i\bullet} = \sum y_{ij}/n_i$$

μ^* and the $\hat{\alpha}_i$'s are calculated from the $\hat{\mu}_i$'s the same way as μ^* and the α_i's are defined in terms of the μ_i's.

For the various definitions of μ^* and α_i

1 $\hat{\mu}^* = \sum\overline{y}_{i\bullet}/g$, $\hat{\alpha}_i = \overline{y}_{i\bullet} - \hat{\mu}^*$

2 $\hat{\mu}^* = \sum n_i\overline{y}_{i\bullet}/N = \sum_i\sum_j y_{ij}/N = \overline{y}_{\bullet\bullet}$

$\hat{\alpha}_i = \overline{y}_{i\bullet} - \overline{y}_{\bullet\bullet}$

3 $\hat{\mu}^* = \overline{y}_{j\bullet}$, $\hat{\alpha}_j = 0$, $\hat{\alpha}_i = \overline{y}_{i\bullet} - \overline{y}_{j\bullet}$, $i \neq j$

Because the estimated effects are different functions of the sample means, their standard errors depend on the definition used.

An <u>Analysis of Variance</u> table consists of

- Several rows, each one associated with one part of the model
- Several columns including some or all of
  - Column to label each row
  - DF (degrees of freedom) column
  - SS (sum of squares) column
  - MS (mean squares) column
  - F-statistic column
  - P-value column corresponding to F's

Here is what a MacAnova ANOVA table looks like:

```
Cmd> anova("logy = treat",fstat:T)
Model used is logy = treat
WARNING: summaries are sequential
           DF         SS           MS            F      P-value
CONSTANT    1     79.425       79.425   8653.95365     < 1e-08
treat       4     3.5376      0.88441     96.36296     < 1e-08
ERROR1     32    0.29369    0.0091779
```

The MS column is SS/DF. The F column are ratios of MS to the error MS. (You won't see P-values like `< 1e-08` until the next release of MacAnova.) `F` and `P-value` are omitted with `fstat:T`.

`anova()` always computes variables **SS** and **DF** matching the sum of squares and degrees of freedom columns. These can be used to check other numbers in the table or compute other quantities depending on them.

```
Cmd> SS
    CONSTANT       treat      ERROR1
      79.425      3.5376     0.29369

Cmd> DF
    CONSTANT       treat      ERROR1
           1           4          32

Cmd> MS <- SS/DF; MS # matches MS column
    CONSTANT       treat      ERROR1
      79.425     0.88441   0.0091779

Cmd> fstats <- MS[-3]/MS[3];fstats
    CONSTANT       treat
        8654      96.363

Cmd> pvalues <- 1 - cumF(fstats,DF[-3], DF[3]); pvalues
(1)            0            0
```


For the completely randomized design, there are potentially three lines corresponding to the three parts of the additive decomposition

$$y_{ij} = \mu^* + \alpha_i + \varepsilon_{ij}$$

Grand mean + treatment effect + error

- μ^* (grand mean)
  $SS = N\overline{y}_{\bullet\bullet}^2$, $DF = 1$.
  Since this line is usually not interesting, many computer programs omit it. MacAnova labels the μ^* line **CONSTANT**.

```
Cmd> grandmean <- describe(logy,mean:T)

Cmd> ssconst <- N*grandmean^2; ssconst
(1)      79.425      Same as in anova() output
```


- α_i's (treatment effects)
  $SS = SS_{trt} = \sum_{1\le i\le g} n_i(\overline{y}_{i\bullet} - \overline{y}_{\bullet\bullet})^2$
  $DF = g - 1$
  If the treatment effects are defined as $\alpha_i = \mu_i - \sum n_i\mu_i/N$, so $\hat{\alpha}_i = \overline{y}_{i\bullet} - \overline{y}_{\bullet\bullet}$, $SS_{trt} = \sum_{1\le i\le g} n_i\hat{\alpha}_i^2$, but this is not true for other definitions, including the one MacAnova uses.

```
Cmd> n <- tabs(logy,treat,n:T) # sample sizes

Cmd> ybars <- tabs(logy,treat,mean:T); ybars # sample means
(1)      1.9325      1.6287      1.3775      1.1943      1.0567

Cmd> sstrt <- sum(n*alphahat^2); sstrt
(1)      3.5376
```

- ε_{ij}'s (errors or residuals)
  $SS = SS_E = \sum_i\sum_j(y_{ij} - \overline{y}_{i\bullet})^2 = \sum_i(n_i - 1)s_i^2$
  $DF = N - g = \sum_{1\le i\le g}(n_i - 1)$

```
Cmd> vars <- tabs(logy,treat,var:T)

Cmd> sse <- sum((n-1)*vars); sse
(1)     0.29369
```


Corresponding to the additive decomposition

$$y_{ij} = \mu^* + \alpha_i + \varepsilon_{ij}$$

there is an additive decomposition of SS

$$\sum_{1\le i\le g}\sum_{1\le j\le n_i} y_{ij}^2 = SS_{const} + SS_{trt} + SS_E$$

associated with μ^* α_i ε_{ij}

Numerical confirmation

```
Cmd> sum(logy^2)
(1,1)      83.256

Cmd> ssconst + sstrt + sse
(1)      83.256
```

And the "total SS"

$$\begin{aligned} SS_T &\equiv \sum_{1\le i\le g}\sum_{1\le j\le n_i}(y_{ij} - \overline{y}_{\bullet\bullet})^2 \\ &= \sum_{1\le i\le g}\sum_{1\le j\le n_i} y_{ij}^2 - N\overline{y}_{\bullet\bullet}^2 \\ &= SS_{const} + SS_{trt} + SS_E - SS_{const} \\ &= SS_{trt} + SS_E \end{aligned}$$

```
Cmd> sst <- sum((logy - grandmean)^2); sst
(1,1)      3.8313     Total SS

Cmd> SS[2] + SS[3] # or sum(SS[-1])
(1)      3.8313      SStrt + SSE
```