Displays for Statistics 5303

Lecture 26

November 4, 2002

Christopher Bingham, Instructor

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

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

New Macro

I have posted a new macro **type2anova()** to compute ANOVA type II SS. You can download it through the class web page

Type II SS:

Type II SS are hierarchical SS. Each SS is the amount of the total SS "explained" by a term after fitting **the largest hierarchical model that does not include the term**.

```
Cmd> data <- read("","exmpl8.10",quiet:T)
Read from file "TP1:Stat5303:Data:OeCh08.dat"

Cmd> makecols(data,assaytemp,growthtemp,variety,activity)

Cmd> assaytemp <- factor(assaytemp)

Cmd> growthtemp <- factor(growthtemp)

Cmd> variety <- factor(variety)

Cmd> logy <- log(activity)

Cmd> logy[1] <- ? # make unbalanced.

Cmd> addmacrofile(getfilename()) # find type2anova.mac
```

getfilename() brings up a file navigation dialog box in which you locate and select the file of macros. Then **addmacrofile()** adds the file to the list of files where MacAnova looks for macros.

```
Cmd> type2anova("logy = (assaytemp+growthtemp+variety)^3",\
     pval:T)
WARNING: searching for unrecognized macro type2anova near
type2anova
                              DF Type II SS         MS      P-value
CONSTANT                       1     3200.5     3200.5            0
assaytemp                      7     3.0375    0.43393            0
growthtemp                     1  0.0020694  0.0020694      0.53521
variety                        1    0.55989    0.55989  4.6629e-15
assaytemp.growthtemp           7   0.067156  0.0095937      0.10245
assaytemp.variety              7   0.026029  0.0037184      0.67307
growthtemp.variety             1   0.078632   0.078632   0.00028496
assaytemp.growthtemp.variety   7   0.053554  0.0076506      0.20654
ERROR1                        63    0.33538  0.0053235          0.5
```

A quick check shows the two-way SS interactions are correct.

```
Cmd> anova("logy = (assaytemp+growthtemp+variety)^2",\
     marginal:T)
Model used is logy = (assaytemp+growthtemp+variety)^2
WARNING: cases with missing values deleted
WARNING: SS are Type III sums of squares
                      DF         SS           MS
CONSTANT               1     3189.9       3189.9
assaytemp              7     3.0241      0.43202
growthtemp             1  0.0030172    0.0030172
variety                1    0.56505      0.56505
assaytemp.growthtemp   7   0.067156    0.0095937
assaytemp.variety      7   0.026029    0.0037184
growthtemp.variety     1   0.078632     0.078632
ERROR1                70    0.38893    0.0055562
```


Two-series Factorial Designs (continued)

I previously entered factors for three replicates of a 2^4 factorial.

```
Cmd> list(A,B,C,D)
A               REAL    64    FACTOR with 2 levels
B               REAL    64    FACTOR with 2 levels
C               REAL    64    FACTOR with 2 levels
D               REAL    64    FACTOR with 2 levels

Cmd> hconcat(A,B,C,D)[run(16),] # first replicate
              (1)         (2)         (3)         (4)
(1)             1           1           1           1
a               2           1           1           1
b               1           2           1           1
ab              2           2           1           1
c               1           1           2           1
ac              2           1           2           1
bc              1           2           2           1
abc             2           2           2           1
d               1           1           1           2
ad              2           1           1           2
bd              1           2           1           2
abd             2           2           1           2
cd              1           1           2           2
acd             2           1           2           2
bcd             1           2           2           2
abcd            2           2           2           2
```

The row labels are the conventional treatment names in *standard order*.

- The presence of a letter indicates the factor was at the high level
- The absence of a letter indicates the factor was at the low level.

I generated a vector of 16 $\mu_{ijk\ell}$'s by treating the factor levels as if they were numerical values. It corresponds to a model of the form

$$\mu_{ijk\ell} = \mu + \alpha_i + \beta_j + \alpha\beta_{ij} + \gamma_k$$

with $\mu = 107$, $\alpha_2 = -\alpha_1 = 0.5$, $\beta_2 = -\beta_1 = 1$, $\gamma_1 = -\gamma_1 = 0.5$.

```
Cmd> mu_ijkl <- 100 + A + 2*B + A*B + C

Cmd> Y <- mu_ijkl + rnorm(64) # artificial data with sigma = 1

Cmd> array(tabs(Y,A,B,C,D,count:T),\
        labels:structure("A","B","C","D")) #sample sizes n_ijkl
                  D1          D2
A1 B1 C1           4           4
      C2           4           4
   B2 C1           4           4
      C2           4           4
A2 B1 C1           4           4
      C2           4           4
   B2 C1           4           4
      C2           4           4
```

These are the $n_{ijk\ell}$.

Statistics 5303 Lecture 26 November 4, 2002

```
Cmd> anova("Y=A*B*C*D")
Model used is Y=A*B*C*D
                DF          SS          MS     P-value
CONSTANT         1  7.4798e+05  7.4798e+05           0
A                1      82.573      82.573  6.1273e-11
B                1      218.43      218.43  4.2375e-18
A.B              1      6.8349      6.8349    0.019937
C                1      10.379      10.379    0.004674
A.C              1      1.6816      1.6816     0.23821
B.C              1     0.77669     0.77669     0.42097
A.B.C            1      1.8476      1.8476     0.21667
D                1     0.25727     0.25727      0.6425
A.D              1    0.031776    0.031776     0.87028
B.D              1      7.0878      7.0878    0.017894
A.B.D            1     0.14331     0.14331     0.72886
C.D              1    0.080634    0.080634      0.7948
A.C.D            1     0.02015     0.02015     0.89653
B.C.D            1      1.7681      1.7681     0.22667
A.B.C.D          1     0.72778     0.72778     0.43589
ERROR1          48      56.585      1.1789

Cmd> effects <- coefs() # get all factorial effects

Cmd> ybar_ijkldot <- array(tabs(Y,A,B,C,D,mean:T),\
       labels:structure("A","B","C","D")); ybar_ijkldot
A1 B1 C1       105.03       105.95   Sample cell means
      C2       105.03       105.79
   B2 C1       108.39       107.55
      C2       109.27       108.75
A2 B1 C1       105.85       107.03
      C2       107.54       107.84
   B2 C1       111.44       110.39
      C2        111.8       112.05

Cmd> ybarvec <- vector(ybar_ijkldot); ybarvec
 (1)      105.03       105.85       108.39       111.44       105.03
 (6)      107.54       109.27        111.8       105.95       107.03
(11)      107.55       110.39       105.79       107.84       108.75
(16)      112.05
```

ybarvec is the vector of $\overline{y}_{ijk\ell\bullet}$ *in standard order*: (1), a, b, ab, c, ac, bc, abc, d, ad, bd, abd, cd, acd, bcd, abcd.

One advantage of 2^k designs is that all effects can be computed by simple contrasts with coefficients ±1.

```
Cmd> concoefs <- hconcat(c_a,c_b,c_ab,c_c,c_ac,c_bc,c_abc,\
  c_d,c_ad,c_bd,c_abd,c_cd,c_acd,c_bcd,c_abcd)

Cmd> print(concoefs,format:"2.0f") # all contrast vectors
concoefs:
 (1,1) -1 -1  1 -1  1  1 -1 -1  1  1 -1  1 -1 -1  1
 (2,1)  1 -1 -1 -1 -1  1  1 -1 -1  1  1  1  1 -1 -1
 (3,1) -1  1 -1 -1  1 -1  1 -1  1 -1  1  1 -1  1 -1
 (4,1)  1  1  1 -1 -1 -1 -1 -1 -1 -1 -1  1  1  1  1
 (5,1) -1 -1  1  1 -1 -1  1 -1  1  1 -1 -1  1  1 -1
 (6,1)  1 -1 -1  1  1 -1 -1 -1 -1  1  1 -1 -1  1  1
 (7,1) -1  1 -1  1 -1  1 -1 -1  1 -1  1 -1  1 -1  1
 (8,1)  1  1  1  1  1  1  1 -1 -1 -1 -1 -1 -1 -1 -1
 (9,1) -1 -1  1 -1  1  1 -1  1 -1 -1  1 -1  1  1 -1
(10,1)  1 -1 -1 -1 -1  1  1  1  1 -1 -1 -1 -1  1  1
(11,1) -1  1 -1 -1  1 -1  1  1 -1  1 -1 -1  1 -1  1
(12,1)  1  1  1 -1 -1 -1 -1  1  1  1  1 -1 -1 -1 -1
(13,1) -1 -1  1  1 -1 -1  1  1 -1 -1  1  1 -1 -1  1
(14,1)  1 -1 -1  1  1 -1 -1  1  1 -1 -1  1  1 -1 -1
(15,1) -1  1 -1  1 -1  1 -1  1 -1  1 -1  1 -1  1 -1
(16,1)  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
```

These 15 contrasts are orthogonal to each other.

```
Cmd> all <- vector(sum(concoefs *ybarvec)) # contrast values

Cmd> all/16 # values of all 15 contrasts divided by 2^k = 2^4
 (1)     1.1359      1.8474      0.3268     0.40271      0.1621
 (6)    0.11016    -0.16991    0.063402    0.022282    -0.33279
(11)   0.047321    0.035495    0.017744     0.16621     0.10664
```

These are all the factorial coefficients $\hat{\alpha}_2$, $\hat{\beta}_2$, $\widehat{\alpha\beta}_{22}$, $\hat{\gamma}_2$, $\widehat{\alpha\gamma}_{22}$, ..., $\widehat{\alpha\beta\gamma\delta}_{2222}$

```
Cmd> vector(effects[2][2],effects[3][2], effects[4][2,2],\
     effects[16][2,2,2,2])
(1)     1.1359      1.8474      0.3268     0.10664
```


Every contrast has the form $sum_+ - sum_-$, where sum_+ is the sum of the $2^{k-1} = 8$ $\bar{y}_{ijkl\bullet}$'s getting weight +1 and sum_- is the sum of the $2^{k-1} = 8$ $\bar{y}_{ijkl\bullet}$'s getting weight -1. So the contrasts divided by 2^{k-1} are a difference of two means. For instance

$$(\sum_{ijk\ell} c_{ijk\ell}{}^A y_{ijk\ell})/8 = \bar{y}_{2\bullet\bullet\bullet} - \bar{y}_{1\bullet\bullet\bullet} = 2\hat{\alpha}_2$$

```
Cmd> all/8 # values of all 15 contrasts divided by 2^(k-1) = 8
 (1)     2.2717      3.6949     0.65359     0.80543     0.32419
 (6)    0.22033    -0.33982      0.1268    0.044565    -0.66558
(11)   0.094642     0.07099    0.035488     0.33242     0.21328

Cmd> means_a <- tabs(Y,A,mean:T);means_a # Factor A means
(1)      106.97      109.24

Cmd> means_a[2] - means_a[1] # 109.24 - 106.97
(1)      2.2717
```

MacAnova function **yates()** is a quick way to compute these from a vector of treatment means in standard order.

```
Cmd> usage(yates)
yates(x), x a REAL vector

Cmd> yates_values <- yates(ybarvec); yates_values
 (1)     2.2717      3.6949     0.65359     0.80543     0.32419
 (6)    0.22033    -0.33982      0.1268    0.044565    -0.66558
(11)   0.094642     0.07099    0.035488     0.33242     0.21328
```

This works only when there are 2^k means.

yates() is an implementation of **Yates' algorithm**, a fairly easy way to compute all the values by hand, without writing down all the contrasts. The algorithm itself is no longer of much interest.

Why do we care about this?

When there is only one replicate, there is no error estimate, and you can be uncertain which terms to pool.

The Yate's effects are helpful in (a) indicating the important effects and (b) giving information about the error.

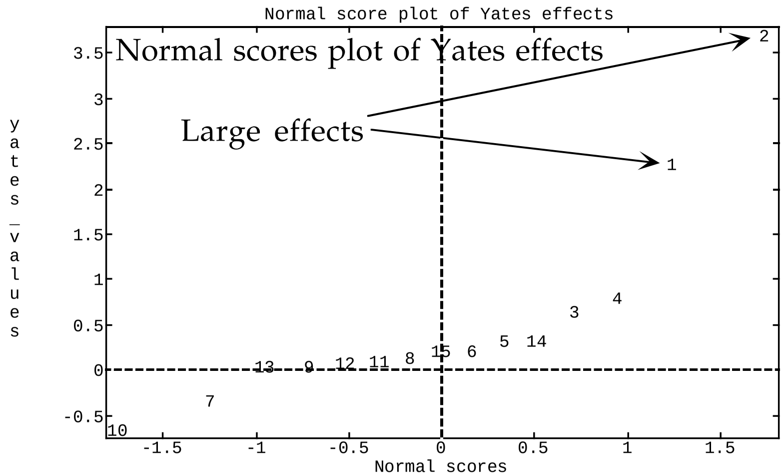
For any terms for which the true effects are 0, the Yates effects behave like independent $N(0,\sigma^2)$.

This suggests making a normal scores plot of the Yates effects, not to test normality, but to spot "outliers", significantly large effects. Large positive effects "stick out" in the upper right hand corner; large negative effects "stick out" in the lower left hand corner.

Alternatively, if you prefer that all the large effects, both positive and negative, "stick out" together, you can make a **"half normal" plot** of $|\text{Yates effects}|$. The half normal scores (computed by **halrnorm()**) are like normal scores for data of the form $|X_i|$, X_i $N(0,\sigma^2)$.

Let's treat the vector of means as if it were a vector from an unreplicated experiment.

```
Cmd> chplot(rankits(yates_values),yates_values,\
     title:"Normal score plot of Yates effects")
```

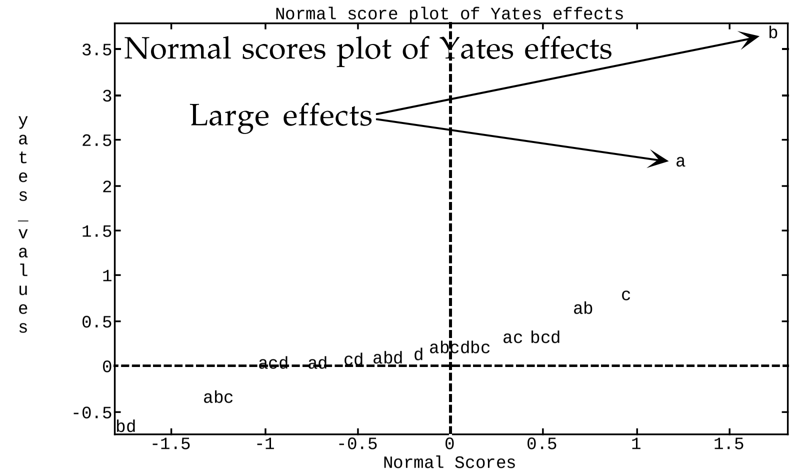


You can use **stringplot()** to make the plot with labels identifying the effect if you want:

```
Cmd> labs <- vector("a","b","ab","c","ac","bc","abc",\
     "d","ad","bd","abd","cd","acd","bcd","abcd")
```

These labels identify terms in the ANOVA, not treatment combinations. It would be more accurate to use, say, **"a.b"** for the AB interaction instead of **"ab"**, but the extra "." would clutter the graph.

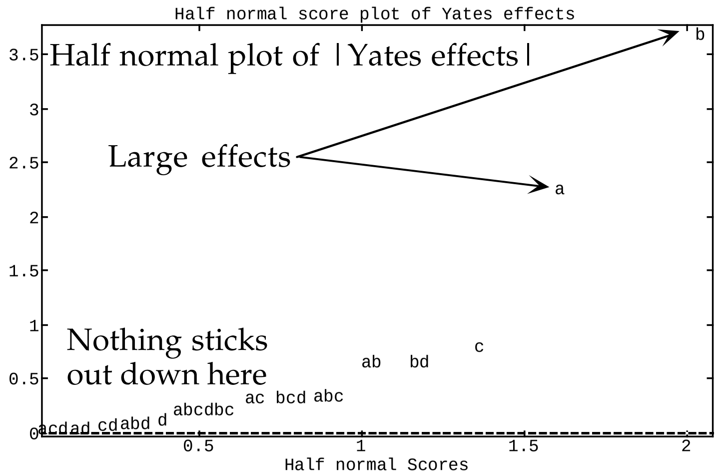

```
Cmd> stringplot(rankits(yates_values),yates_values,labs,\
     title:"Normal score plot of Yates effects",\
     xlab:"Normal Scores")
```



Because of the way the data were generated, you know the only non-zero true effects are for A, B, AB and C.

Even though the C and AB effects don't stick out as A and B do, they are the next in order of size.

```
Cmd> stringplot(halfnorm(abs(yates_values)),abs(yates_values),\
     labs,xlab:"Half normal Scores",\
     title:"Half normal score plot of Yates effects")
```



A more formal procedure to identify large effects is due to Lenth. He proposed using a PSE = pseudo standard error.

Let s_0 = 1.5×median(|yates effects|)

Then PSE =
  1.5×median(|yates effects| < $2.5s_0$)

is a crude estimate of SE(Yates effect)

Then treat (yates effects)/PSE like t-statistics on $(2^k - 1)/3$ DF.


```
Cmd> s_0 <- 1.5*describe(abs(yates_values),median:T)

Cmd> J <- abs(yates_values) <= 2.5*s_0 # T except for a and b

Cmd> labs[!J] # effects deleted in computing PSE
(1) "a"
(2) "b"

Cmd> pse <- 1.5*describe(abs(yates_values)[J],median:T);pse
(1)    0.33049   Pseudo standard error

Cmd> t_stats <- yates_values/pse;t_stats
 (1)      6.8739       11.18      1.9777      2.4371     0.98094
 (6)     0.66667     -1.0282     0.38369     0.13485     -2.0139
(11)     0.28637      0.2148     0.10738      1.0059     0.64533
```

Only first two, A and B, are large

```
Cmd> df <- (2^4 - 1)/3; df
(1)           5

Cmd> twotailt(t_stats,df) # ordinary P-values
 (1)  0.00099661    9.99e-05     0.10489    0.058864     0.37167
 (6)     0.53451     0.35099     0.71699     0.89799     0.10014
(11)     0.78608     0.83841     0.91866     0.36065     0.54715

Cmd> 15*twotailt(t_stats,df)
 (1)    0.014949   0.0014985      1.5734     0.88296      5.5751
 (6)      8.0176      5.2648      10.755       13.47      1.5022
(11)      11.791      12.576       13.78      5.4098      8.2073
```

Only A and B are significant. Neither AB interaction or C main effect was.

This suggests you may be everything except A and B. But since AB and C are next you probably should check their F-statistics.

To use **anova()** on on the vector means **ybarvec** you need factor vectors for one replicate. You can get these from the first 16 rows of factors **A**, **B** and **C**.

```
Cmd> A1 <- factor(A[run(16)]);B1 <- factor(B[run(16)])

Cmd> C1 <- factor(C[run(16)]);D1 <- factor(D[run(16)])

Cmd> anova("ybarvec = A1*B1 + C1",pvals:T)
Model used is ybarvec = A1*B1 + C1
                DF          SS          MS     P-value
CONSTANT         1  1.8699e+05  1.8699e+05  2.7525e-27
A1               1      20.643      20.643  7.0516e-06
B1               1      54.608      54.608  5.4814e-08
A1.B1            1      1.7087      1.7087    0.043294
C1               1      2.5949      2.5949     0.01686
ERROR1          11      3.6057     0.32779
```

In this analysis, both AB and C are nominally significant. But when you take into account the number of terms that were screened, you would certainly not have a lot of confidence the effects were real. (Of course, you know they are real because you know how the data were generated. In the real world you don't.)

Random and fixed effects

In the ANOVA models we have so far considered, each $y_{ij...}$ has just one random part $\varepsilon_{ij...}$. We have considered everything else to be non-random or **fixed parameters**.

- One factor:

  $y_{ij} = \mu_i + \varepsilon_{ij} = \mu + \alpha_i + \varepsilon_{ij}$

  μ, $\{\alpha_i\}$ are not random but **fixed**

- Two factors:

  $y_{ijk} = \mu_{ij} + \varepsilon_{ij} = \mu + \alpha_i + \beta_k + \alpha\beta_{ij} + \varepsilon_{ij}$

  μ, $\{\alpha_i\}$, $\{\beta_j\}$, $\{\alpha\beta_{ij}\}$ are **fixed**

Inference has mainly been about the values of the fixed parameters.

- Testing H_0: $\alpha_1 = \alpha_2 = \ldots = \alpha_a = 0$
- Testing H_0: All $\alpha\beta_{ij} = 0$
- Finding which α_i's differ from which (factor effects multiple comparisons).

In many situations it is not realistic or at least not sensible to view the factor effects as fixed.

Oehlert example:

A company has 50 machines to make cardboard cartons and they want to understand the sources of the variation in strength of cartons they produce.

They choose 10 machines randomly and make 40 cartons on each machine, 400 in all, making each box from a different lot of cardboard.

This looks like a one-factor experiment with $a = 10$ machines as factor levels and $n_1 = n_2 = \ldots = n_{10} = 40$ replicates of each factor level.

But in the factorial model

$$y_{ij} = \mu + \alpha_i + \varepsilon_{ij},\ i = 1,\ldots,a, j = 1,\ldots,n_i$$

the machine effects α_i's are should probably be thought of as random. This is true, even for each individual machine $\alpha_i = \mu_i - \mu$ is fixed, because the machines were selected randomly. So the unobserved α_i are a random sample from a population of 50 possible values.

μ has a different interpretation from μ in a fixed effects model where μ is the average of the means for each factor level, that is $\mu = \overline{\mu}_{\bullet}$. Instead

$$\mu = E(\text{average strength on a random machine})$$

that is, an average of *all 50 means*, not not the average of the 10 means μ_i that happened to be sampled.

$\alpha_i = \mu_i - \mu$ is the deviation of the mean for the i^{th} machine sampled from the population mean μ and is a random variable with $E(\alpha_i) = 0$.

In random effect models, the property $E(\alpha_i) = 0$ takes the place of the the restriction $\sum_i \alpha_i = 0$. Indeed it is very likely $\sum_i \alpha_i \neq 0$.

There are really only three parameters that might be said to be fixed.

- μ = average all the μ_i's over the entire population, not the sample.
- $\sigma_\alpha^2 = Var(\alpha_i)$ = variance of machine effect = between machine variance.
- $\sigma^2 = Var(\varepsilon_{ij})$ = variance of box strength among boxes made on the same machine = within machine variance.

μ, σ_α^2 and σ^2 are at the focus of statistical inference for random effect models. In more complicated designs there can be many more variances.