Displays for Statistics 5303

Lecture 25

November 1, 2002

Christopher Bingham, Instructor

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

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

Types of sums of squares

I need to make a correction to what I said in Lecture 24 about Type II SS.

The definition I gave was correct; I applied it incorrectly.

SAS **Type I SS** (no change)

Type I SS are **sequential SS** like MacAnova. Each SS is the amount of the total SS "explained" by that term *after* fitting *preceding* terms.

In MacAnova, the terms that precede any term never "contain" it, in the sense that AB contains A and B and ABC contains A, B, C, AB, AC, BC.

Such non-hierarchical orders may be possible in other programs. In R, you can put B after A and AB but R treats the model as if the terms were A, B, AB in that order.

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.**

A is tested in the model

$$y = \mu + \alpha_i + \beta_j + \gamma_k + \beta\gamma_{jk} + \varepsilon_{ijk}$$

None of $\alpha\gamma_{ik}$, $\alpha\beta_{ij}$ or $\alpha\beta\gamma_{ijk}$ is included since they all "contain" α.

B is tested in the model

$$y = \mu + \alpha_i + \beta_j + \gamma_k + \alpha\gamma_{ik} + \varepsilon_{ijk}$$

C is tested in the model

$$y = \mu + \alpha_i + \beta_j + \gamma_k + \alpha\beta_{ij} + \varepsilon_{ijk}$$

AB, AC and BC are tested in the model

$$y = \mu + \alpha_i + \beta_j + \gamma_k + \alpha\beta_{ij} + \alpha\gamma_{ik} + \beta\gamma_{jk} + \varepsilon_{ijk}$$

$\alpha\beta\gamma_{ijk}$ isn't included since it "contains" each two-way interaction.

For the full three-way interaction model as specified by `"y=(a+b+c)^3"`, say, the type II SS are

SS(A | 1,B,C,BC),
SS(B | 1,A,C,AC),
SS(C | 1,A,B,AB),
SS(AB | 1,A,B,C,AC,BC),
SS(AC | 1,A,B,C,AB,BC),
SS(BC | 1,A,B,C,AB,AC)
SS(ABC | 1,A,B,C,AB,AC,BC)

Type III SS

Each SS is the SS "explained" by the term after fitting *all* the other terms in the model.

So, for example, in fitting the full three-way interaction model

SS_A = SS(A | 1,B,C,AB,AC,BC,ABC)

SS_{AB} = SS(AB | 1,A,B,C,AC,BC,ABC)

etc.

The type III SS_A in a 3-factor model with 3-way interaction is the SS to test

$$H_0: \alpha_1 = \alpha_2 = \ldots = \alpha_a$$

in fitting all the coefficients in

$$\mu_{ijk} = \mu+\alpha_i+\beta_j+\gamma_k+\alpha\beta_{ij}+\alpha\gamma_{ik}+\beta\gamma_{jk}+\alpha\beta\gamma_{ijk}$$

In the context of this model H_0 is

$$H_0: \mu_{1\bullet\bullet} = \mu_{2\bullet\bullet} = \ldots = \mu_{a\bullet\bullet}$$

where $\mu_{i\bullet\bullet} = (1/bc)\sum_j\sum_k\mu_{ijk}$ is the average of all μ_{ijk} with first subscript i.

The Type III SS_{AB} similarly tests $H_0: \alpha\beta_{ij} = 0$, all i and j, in fitting all the coefficients in

$$\mu_{ijk} = \mu+\alpha_i+\beta_j+\gamma_k+\alpha\beta_{ij}+\alpha\gamma_{ik}+\beta\gamma_{jk}+\alpha\beta\gamma_{ijk}$$

H_0 is the same as H_0: All $\mu_{ij\bullet}-\mu_{i\bullet\bullet}-\mu_{\bullet j\bullet}+\mu_{\bullet\bullet\bullet}$ are equal, where $\mu_{ij\bullet} = (1/c)\sum_k\mu_{ijk}$.

Comments:

- Type III SS_A, SS_B and SS_C for the 3-factor main effect model $\mu_{ijk} = \alpha_i + \beta_j + \gamma_k$ are *not* type II SS when fitting two- and three-way interaction models (this corrects Lecture 24)
- Type III SS_{AB}, SS_{AC}, SS_{BC} for the 3-factor two-way interaction model $\mu_{ijk} = \alpha_i + \beta_j + \gamma_k + \alpha\beta_{ij} + \alpha\gamma_{ik} + \beta\gamma_{jk}$ *are* also type II SS when fitting two- and three-way interaction models.

You can get all type II SS for a 3-factor ANOVA from only 3 type I (sequential) ANOVAs.

You can get all type II two-way interactions from type III SS from fitting one ANOVA with no three way interaction (`anova("y=(a+b+c)^2",marginal:T)`).

Using shortcut `logy=(assaytemp + growthtemp + variety)^3` to express the complete three way model is not the way to go to get a lot of Type II SS.

```
Cmd> anova("logy=(assaytemp + growthtemp + variety)^3")
Model used is logy=(assaytemp + growthtemp + variety)^3
WARNING: cases with missing values deleted
WARNING: summaries are sequential
                              DF          SS          MS
CONSTANT                       1      3200.5      3200.5
assaytemp                      7      3.0628     0.43755
growthtemp                     1    0.001396    0.001396
variety                        1     0.55282     0.55282
assaytemp.growthtemp           7     0.06407   0.0091529
assaytemp.variety              7    0.025892   0.0036989
growthtemp.variety             1    0.078632    0.078632
assaytemp.growthtemp.variety   7    0.053554   0.0076506
ERROR1                        63     0.33538   0.0053235
```

Because `marginal:T` was not an argument, all SS all type I.

The underlined terms are also type II SS but no main effect SS are type II SS.

Short cut `logy=assaytemp*growthtemp*variety` is more productive of type II SS. It produces one Type II main effect SS, one type II two-way interaction SS and one type II three-way interaction SS.

When you are fitting an additive (main effect model), you can get all SS from one ANOVA using `marginal:T`. In this model the main effect type III SS are also type II SS because there are no interactions.

`marginal:T` with main effect model

```
Cmd> anova("logy=assaytemp + growthtemp + variety",marginal:T)
Model used is logy=assaytemp + growthtemp + variety
WARNING: cases with missing values deleted
WARNING: SS are Type III sums of squares
                DF          SS          MS
CONSTANT         1      3196.6      3196.6
assaytemp        7       3.044     0.43486
growthtemp       1   0.0021074   0.0021074
variety          1     0.55282     0.55282
ERROR1          85     0.55753   0.0065591
```

The order of terms doesn't matter:

```
Cmd> anova("logy=variety + growthtemp + assaytemp",marginal:T)
Model used is logy=variety + growthtemp + assaytemp
WARNING: cases with missing values deleted
WARNING: SS are Type III sums of squares
                DF          SS          MS
CONSTANT         1      3196.6      3196.6
variety          1     0.55282     0.55282
growthtemp       1   0.0021074   0.0021074
assaytemp        7       3.044     0.43486
ERROR1          85     0.55753   0.0065591
```

To get these SS without using **marginal:T**, you would have to do three ANOVAS, one with each of the terms last.

For example, this one has **growthtemp** last and $SS_{\text{growthtemp}}$ matches the Type III SS just computed.

```
Cmd> anova("logy=variety+assaytemp+growthtemp")
Model used is logy=growthtemp+variety+assaytemp
WARNING: cases with missing values deleted
WARNING: summaries are sequential
                DF          SS          MS
CONSTANT         1      3200.5      3200.5
variety          1     0.56975     0.56975
assaytemp        7      3.0452     0.43503
growthtemp       1   0.0021074   0.0021074
ERROR1          85     0.55753   0.0065591
```


To get all Type II SS with a two- or three-factor model you need three sequential ANOVAs without **marginal:T**

Order **variety, assaytemp, growthtemp**:

```
Cmd> anova("logy=variety*assaytemp*growthtemp")
Model used is logy=variety*assaytemp*growthtemp
WARNING: cases with missing values deleted
WARNING: summaries are sequential
                                DF          SS          MS
CONSTANT                         1      3200.5      3200.5
variety                          1     0.56975     0.56975
assaytemp                        7      3.0452     0.43503
variety.assaytemp                7    0.026078   0.0037254
growthtemp                       1   0.0020694   0.0020694
variety.growthtemp               1    0.075399    0.075399
assaytemp.growthtemp             7    0.067156   0.0095937
variety.assaytemp.growthtemp     7    0.053554   0.0076506
ERROR1                          63     0.33538   0.0053235
```

The 3 underlined terms are Type II SS.

Order **growthtemp, variety, assaytemp**:

```
Cmd> anova("logy=growthtemp*variety*assaytemp")
Model used is logy=growthtemp*variety*assaytemp
WARNING: cases with missing values deleted
WARNING: summaries are sequential
                                DF          SS          MS
CONSTANT                         1      3200.5      3200.5
growthtemp                       1   0.0024441   0.0024441
variety                          1     0.57061     0.57061
growthtemp.variety               1     0.08202     0.08202
assaytemp                        7      3.0375     0.43393
growthtemp.assaytemp             7    0.067028   0.0095754
variety.assaytemp                7    0.026029   0.0037184
growthtemp.variety.assaytemp     7    0.053554   0.0076506
ERROR1                          63     0.33538   0.0053235
```

The 3 underlined terms are Type II SS.

Order `assaytemp, growthtemp, variety`:

```
Cmd> anova("logy=assaytemp*growthtemp*variety")
Model used is logy=assaytemp*growthtemp*variety
WARNING: cases with missing values deleted
WARNING: summaries are sequential
                              DF          SS          MS
CONSTANT                       1      3200.5      3200.5
assaytemp                      7      3.0628     0.43755
growthtemp                     1    0.001396    0.001396
assaytemp.growthtemp           7    0.056997   0.0081425
variety                        1     0.55989     0.55989
assaytemp.variety              7    0.025892   0.0036989
growthtemp.variety             1    0.078632    0.078632
assaytemp.growthtemp.variety   7    0.053554   0.0076506
ERROR1                        63     0.33538   0.0053235
```

You can also get the Type II two-way interaction SS from one ANOVA with `marginal:T` without an ABC term.

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

but you can't get the Type II main effects using `marginal:T` when all the interactions are in the model

For 4 factors, you need 6 ANOVA with models

`"y=a*b*c*d"`	SS_D, SS_{CD}, SS_{BCD}, SS_{ABCD}
`"y=d*a*b*c"`	SS_C, SS_{BC}, SS_{ABC}, (SS_{ABCD})
`"y=c*d*a*b"`	SS_B, SS_{AB}, SS_{ABD}, (SS_{ABCD})
`"y=b*c*d*a"`	SS_A, SS_{AD}, SS_{ACD}, (SS_{ABCD})
`"y=a*c*d*b"`	(SS_B), SS_{BD}, (SS_{BCD}), (SS_{ABCD})
`"y=b*d*c*a"`	(SS_A), SS_{AC}, (SS_{ACD}), (SS_{ABCD})

Terms in parentheses are found more than once.

```
Cmd> anova("logy=variety*assaytemp*growthtemp",marginal:T)
Model used is logy=variety*assaytemp*growthtemp
WARNING: cases with missing values deleted
WARNING: SS are Type III sums of squares
                              DF          SS          MS
CONSTANT                       1      3184.6      3184.6
variety                        1     0.55812     0.55812
assaytemp                      7      3.0304     0.43292
variety.assaytemp              7    0.025891   0.0036987
growthtemp                     1   0.0025845   0.0025845
variety.growthtemp             1     0.07626     0.07626
assaytemp.growthtemp           7    0.063516   0.0090737
variety.assaytemp.growthtemp   7    0.053554   0.0076506
ERROR1                        63     0.33538   0.0053235
```

These are the full Type III SS. Only SS_{ABC} matches the original Type I SS computed without `marginal:T`.

Two-series Factorial Designs

Notation: A design with k_a factors with a levels, k_b factors with b levels, ..., k_z factors with z levels, is a $a^{k_a}b^{k_b}...z^{k_z}$ design. For example, a design with 3 factors with levels 2, 2 and 3 is a $2^2 3^1$ design.

2^k designs with k factors, each with only two levels, are particularly important

Levels 1 and 2 of each factor are traditionally called the "low" and "high" levels.

Letters a, b, c, ... are traditionally used to designate the high level of factors A, B, C, ...,.

Notation

The treatment combination of low levels of all factors is notated (1).

All other treatment combinations are notated with a combination of the letters a, b, c, ..., with the letter present if the corresponding factor is at the high level.

	A	B	C
(1)	1	1	1
a	2	1	1
b	1	2	1
ab	2	2	1
c	1	1	2
ac	2	1	2
bc	1	2	2
abc	2	2	2

This is an example of **standard order**.

k = 1 (1), a

k = 2 (1), a, b, ab

k = 3 (1), a, b, ab, c, ac, bc, abc

k = 4 (1), a, b, ab, c, ac, bc, abc
d, ad, bd, abd, cd, acd, bcd, abcd

The first 2^{k-1} combinations match the 2^{k-1} design. You get the rest by combining the 2^{k-1} combinations with the new letter.

```
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),]
           (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
```


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

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

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

Cmd> Y <- mu + rnorm(64) # artificial data with signma = 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
```

Type **help(labels)** for information on adding labels to data.

All sample sizes are the same so the data are **balanced**.

```
Cmd> anova("Y=A*B*C*D")
Model used is Y=A*B*C*D
                DF          SS          MS
CONSTANT         1  7.4798e+05  7.4798e+05
A                1      82.573      82.573
B                1      218.43      218.43
A.B              1      6.8349      6.8349
C                1      10.379      10.379
A.C              1      1.6816      1.6816
B.C              1     0.77669     0.77669
A.B.C            1      1.8476      1.8476
D                1     0.25727     0.25727
A.D              1    0.031776    0.031776
B.D              1      7.0878      7.0878
A.B.D            1     0.14331     0.14331
C.D              1    0.080634    0.080634
A.C.D            1     0.02015     0.02015
B.C.D            1      1.7681      1.7681
A.B.C.D          1     0.72778     0.72778
ERROR1          48      56.585      1.1789
```

Note the terms are in "standard" order with **CONSTANT** in place of (1).

```
Cmd> ybar_ijkldot <- array(tabs(Y,A,B,C,D,mean:T),\
     labels:structure("A","B","C","D")); ybar_ijkldot
                  D1          D2
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.

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

Create factorial contrasts

```
Cmd> c_a <- vector(-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1)

Cmd> c_b <- vector(-1,-1,1,1,-1,-1,1,1,-1,-1,1,1,-1,-1,1,1)

Cmd> c_c <- vector(-1,-1,-1,-1,1,1,1,1,-1,-1,-1,-1,1,1,1,1)

Cmd> c_d <- vector(-1,-1,-1,-1,-1,-1,-1,-1,1,1,1,1,1,1,1,1)

Cmd> print(hconcat(c_a,c_b,c_c,c_d),format:"3.0f")
MATRIX:
 (1,1)  -1  -1  -1  -1
 (2,1)   1  -1  -1  -1
 (3,1)  -1   1  -1  -1
 (4,1)   1   1  -1  -1
 (5,1)  -1  -1   1  -1
 (6,1)   1  -1   1  -1
 (7,1)  -1   1   1  -1
 (8,1)   1   1   1  -1
 (9,1)  -1  -1  -1   1
(10,1)   1  -1  -1   1
(11,1)  -1   1  -1   1
(12,1)   1   1  -1   1
(13,1)  -1  -1   1   1
(14,1)   1  -1   1   1
(15,1)  -1   1   1   1
(16,1)   1   1   1   1
```

You can calculate the contrast for factor j of k by **rep(rep(run(2), rep(2^(j-1),2)),2^(k-j))**

```
Cmd> k <- 4;j <- 3

Cmd> print(rep(rep(run(2),rep(2^(j-1),2)),\
     2^(k-j)),format:"2.0f")
VECTOR:
(1)  1  1  1  1  2  2  2  2  1  1  1  1  2  2  2  2
```

Get structure with all factorial effects

```
Cmd> effects <- coefs()

Cmd> compnames(effects) # names of components
 (1) "CONSTANT"
 (2) "A"
 (3) "B"
 (4) "A.B"
 (5) "C"
 (6) "A.C"
 (7) "B.C"
 (8) "A.B.C"
 (9) "D"
(10) "A.D"
(11) "B.D"
(12) "A.B.D"
(13) "C.D"
(14) "A.C.D"
(15) "B.C.D"
(16) "A.B.C.D"

Cmd> effects$A # alpha_hat
(1)     -1.1359      1.1359

Cmd> sum(c_a*ybarvec)/16 # same as alpha_hat[2]
(1)      1.1359
```

$$\hat{\alpha}_2 = -\hat{\alpha}_1 = \sum_i\sum_j\sum_k c_{ijk}\,\overline{{}^{A}y}_{ijk\bullet}/16$$

```
Cmd> vector(sum(c_b*ybarvec)/16,effects$B[2])
(1)      1.8474      1.8474 beta_hat[2]
```

$$\hat{\beta}_2 = -\hat{\beta}_1 = \sum_i\sum_j\sum_k c_{ijk}\,\overline{{}^{B}y}_{ijk\bullet}/16$$

```
Cmd> vector(sum(c_c*ybarvec)/16,effects$C[2])
(1)     0.40271     0.40271 gamma_hat[2]

Cmd> vector(sum(c_d*ybarvec)/16,effects$D[2])
(1)    0.063402    0.063402 delta_hat[2]
```

Thus the high level coefficients are the contrasts divided by 16 = 2^k.

Compute interaction contrasts as products of main effect contrasts.

```
Cmd> c_ab <- c_a*c_b; c_ac <- c_a*c_c; c_ad <- c_a*c_d

Cmd> c_bc <- c_b*c_c; c_bd <- c_b*c_d; c_cd <- c_c*c_d

Cmd> c_abc <- c_a*c_b*c_c; c_abd <- c_a*c_b*c_d

Cmd> c_acd <- c_a*c_c*c_d; c_bcd <- c_b*c_c*c_d

Cmd> c_abcd <- c_a*c_b*c_c*c_d

Cmd> vector(sum(c_ab*ybarvec)/16,effects[4][2,2])
(1)      0.3268      0.3268 alphabetahat[2,2]
```

$$\hat{\alpha\beta}_{22} = -\hat{\alpha\beta}_{12} = -\hat{\alpha\beta}_{12} = \hat{\alpha\beta}_{11} = \sum_i\sum_j\sum_k c_{ijk}\,\overline{{}^{A}y}_{ijk\bullet}/16$$

Use the term names to select effects

```
Cmd> names <- TERMNAMES[-17];print(paste(names))
CONSTANT A B A.B C A.C B.C A.B.C D A.D B.D A.B.D C.D A.C.D B.C.D
A.B.C.D
Cmd> vector(sum(c_ac*ybarvec)/16,effects[names=="A.C"][2,2])
(1)      0.1621      0.1621 alphagammahat[2,2]

Cmd> vector(sum(c_ad*ybarvec)/16,effects[names=="A.D"][2,2])
(1)    0.022282    0.022282 alphadeltahat[2,2]

Cmd> vector(sum(c_abcd*ybarvec)/16,\
    effects[names=="A.B.C.D"][2,2,2,2])
(1)     0.10664     0.10664 alphabetagammadelta[2,2]
```

In each case the factorial coefficient associated with the high levels of all factors in the term is the contrast value divided by 16 = 2^k.

```
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

Cmd> all <- vector(sum(concoefs *ybarvec))

Cmd> all/8 # values of all 15 contrasts divided by 2^(k-1)
 (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
```

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(ybarvec)
 (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
```