

Skeleton ANOVA tables are important for testing and estimation.

One factor skeleton table

Source	DF	EMS
Treatments	a-1	$\sigma^2 + n\sigma_\alpha^2$
Error	N-a	$\sigma^2 = \sigma_\epsilon^2$

Displays for Statistics 5303
 Lecture 28
 November 8, 2002

Christopher Bingham, Instructor

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

Class Web Page

<http://www.stat.umn.edu/~kbc/classes/5303>

© 2002 by Christopher Bingham

Two factor skeleton table

Source	DF	EMS
A	a-1	$\sigma^2 + n\sigma_{\alpha\beta}^2 + nb\sigma_\alpha^2$
B	b-1	$\sigma^2 + n\sigma_{\alpha\beta}^2 + na\sigma_\beta^2$
AB	(a-1)(b-1)	$\sigma^2 + n\sigma_{\alpha\beta}^2$
Error	ab(n-1)	σ^2

The multiplier of a term is the number of cases affected by one effect of that type

- 1 case is affected by each ϵ_{ijk}
- n cases are affected by each $\alpha\beta_{ij}$
- nb cases are affected by each α_i
- na cases are affected by each β_j

For the box-making machines, a = 10, b = 10, n = 4 so the table is

Source	DF	EMS
A:Machines	9	$\sigma^2 + 4\sigma_{\alpha\beta}^2 + 40\sigma_{\alpha}^2$
B:Operators	9	$\sigma^2 + 4\sigma_{\alpha\beta}^2 + 40\sigma_{\beta}^2$
AB	81	$\sigma^2 + 4\sigma_{\alpha\beta}^2$
Error	300	σ^2

Note that $EMS_A = EMS_{AB} + 40\sigma_{\alpha}^2$.

This means that $EMS_A = EMS_{AB}$ if and only if $\sigma_{\alpha}^2 = 0$.

F = MS_1/MS_2 really tests $H_0: E(MS_1) = E(MS_2)$. So to test $H_0: \sigma_{\alpha}^2 = 0$ the proper F-statistic is $F = MS_A/MS_{AB}$ (denominator = MS_{AB}).

This is *different* from the fixed effect case where you use $F = MS_A/MS_E$ (denominator = MS_{error}) to test $H_0: \text{all } \alpha_i = 0$.

Three factor skeleton table

Source	DF	EMS
A	a-1	$\sigma^2 + n\sigma_{\alpha\beta\gamma}^2 + nc\sigma_{\alpha\beta}^2 + nb\sigma_{\alpha\gamma}^2 + nbc\sigma_{\alpha}^2$
B	b-1	$\sigma^2 + n\sigma_{\alpha\beta\gamma}^2 + nc\sigma_{\alpha\beta}^2 + na\sigma_{\beta\gamma}^2 + nacc\sigma_{\beta}^2$
C	c-1	$\sigma^2 + n\sigma_{\alpha\beta\gamma}^2 + nb\sigma_{\alpha\gamma}^2 + na\sigma_{\beta\gamma}^2 + nab\sigma_{\gamma}^2$
AB	(a-1)(b-1)	$\sigma^2 + n\sigma_{\alpha\beta\gamma}^2 + nc\sigma_{\alpha\beta}^2$
AC	(a-1)(c-1)	$\sigma^2 + n\sigma_{\alpha\beta\gamma}^2 + nb\sigma_{\alpha\gamma}^2$
BC	(b-1)(c-1)	$\sigma^2 + n\sigma_{\alpha\beta\gamma}^2 + na\sigma_{\beta\gamma}^2$
ABC	(a-b)(b-1)(c-1)	$\sigma^2 + n\sigma_{\alpha\beta\gamma}^2$
Error	abc(n-1)	σ^2

- n cases affected by each $\alpha\beta\gamma_{ijk}$
- nc cases affected by each $\alpha\beta_{ij}$
- nbc cases affected by each α_i , etc.

Here's a an example of a balanced one factor random effect experiment. The data are weights of calves sired by a = 5 bulls, n = 8 calves per bull

```
Cmd> sire <- factor(1,1,1,1,1,1,1,1, 2,2,2,2,2,2,2,2,
3,3,3,3,3,3,3,3, 4,4,4,4,4,4,4,4, 5,5,5,5,5,5,5,5)
Cmd> wts <- vector(61,100,56,113,99,103,75,62,\
75,102,95,103,98,115,98,94, 58,60,60,57,57,59,54,100,\
57,56,67,59,58,121,101,101, 59,46,120,115,115,93,105,75)
Cmd> anova("wts=sire", fstat:"F")
Model used is wts=sire
```

	DF	SS	MS	F	P-value
CONSTANT	1	2.7258e+05	2.7258e+05	587.71949	0
sire	4	5591.1	1397.8	3.01382	0.030874
ERROR1	35	16233	463.79		

The interest here is the contribution to the variability of weights due to parent.

ems() computes EMS formulas

```
Cmd> ems("wts=sire", "sire")
EMS(CONSTANT) = V(ERROR1) + 8V(sire) + 40Q(CONSTANT)
EMS(sire) = V(ERROR1) + 8V(sire)
EMS(ERROR1) = V(ERROR1)
```

V(ERROR1) stands for σ^2 .

V(sire) stands for σ_α^2

Q(CONSTANT) stands for μ^2 , a function of the fixed parameter μ

From the output

$$\begin{aligned} \text{EMS}_{\text{constant}} &= \sigma^2 + 8\sigma_\alpha^2 + 40\mu^2 \\ \text{EMS}_A &= \sigma^2 + 8\sigma_\alpha^2 \end{aligned}$$

The multipliers here are n = 8 and n_a = 40.

If there was any reason to test $H_0: \mu = 0$ (there isn't in this case), the formulas show you that the proper F-statistic would be $F = \text{MS}_{\text{constant}} / \text{MS}_A$ on 1 and 4 d.f.

```
Cmd> 2.7258e+05/1397.8
(1) 195.01
Cmd> 1 - cumF(195.01,1,4)
(1) 0.00015252
```

When data are unbalanced, the formulas are harder but can be computed by `ems()`. Here I set 4 responses to MISSING and ran `ems()` again.

```
Cmd> wts1 <- wts; wts1[vector(2, 11, 12, 29,30)] <- ?
Cmd> tabs(wts1,sire,count:T) # it's now unbalanced
WARNING: MISSING values in argument 1 to tabs() omitted
(1)
      7      6      8      6      8
Cmd> ems("wts1=sire", "sire")
EMS(CONSTANT) = V(ERROR1) + 7.1143V(sire) + 35Q(CONSTANT)
EMS(sire) = V(ERROR1) + 6.9714V(sire)
EMS(ERROR1) = V(ERROR1)
```

$$EMS_{sire} = \sigma^2 + 6.9714\sigma_\alpha^2$$

This tells you that $F = MS_A / MS_{error}$ is still OK for testing $H_0: \sigma_\alpha^2 = 0$.

But $F = MS_{const} / MS_A$ is no longer OK to test $\mu = 0$, since

$$EMS_{constant} - EMS_A = 35\mu^2 + 0.1429\sigma_A^2$$

Once you get beyond two-way designs, testing gets more complicated.

Suppose you want to test $H_0: \sigma_\alpha^2 = 0$:

$$EMS_A = \sigma^2 + n\sigma_{\alpha\beta\gamma}^2 + nc\sigma_{\alpha\beta}^2 + nb\sigma_{\alpha\gamma}^2 + nbcc\sigma_\alpha^2$$

When H_0 is true,

$$EMS_A = \sigma^2 + n\sigma_{\alpha\beta\gamma}^2 + nc\sigma_{\alpha\beta}^2 + nb\sigma_{\alpha\gamma}^2$$

but there is no term with this EMS to use as a denominator MS in an F-statistic. You need to find a numerator and denominator MS such that

$$E(MS_{num}) - E(MS_{den}) = const \times \sigma_\alpha^2$$

so that when you compare MS_{num} and MS_{den} using $F = MS_{num} / MS_{den}$ you are comparing two quantities whose means are the same when H_0 is true.

One approach (not a good one, but a natural one):

Include both MS_{AB} and MS_{AC} in the denominator so that EMS contains both $n\sigma_{\alpha\beta}^2$ and $nb\sigma_{\alpha\delta}^2$. Since the EMS now includes $2 \times n\sigma_{\alpha\beta\delta}^2$ and EMS_A has only $n\sigma_{\alpha\beta\delta}^2$, also subtract MS_{ABC} to get rid of the extra $n\sigma_{\alpha\beta\delta}^2$. This leads to

$$F = MS_A / (MS_{AB} + MS_{AC} - MS_{ABC}).$$

- **Advantage:** $MS_{num} = MS_A$, the fixed effects numerator.
- **Disadvantage:** It's possible to have $MS_{den} < 0$ and hence $F < 0$ which can never happen with a real F-statistic.

The better approach is to find MS_{num} and MS_{den} using only positive coefficients.

Approach using positive coefficients

Include MA_{ABC} in MS_{num} to compensate for the extra $n\sigma_{\alpha\beta}^2$ in $E(MS_A + MS_{ABC})$.

$$F = (MS_A + MS_{ABC}) / (MS_{AB} + MS_{AC})$$

$$\begin{aligned} E(MS_{denom}) &= E(MS_{AB} + MS_{AC}) \\ &= 2\sigma^2 + 2n\sigma_{\alpha\beta\delta}^2 + n\sigma_{\alpha\beta}^2 + nb\sigma_{\alpha\delta}^2 \\ E(MS_{num}) &= E(MS_A + MS_{ABC}) \\ &= 2\sigma^2 + 2n\sigma_{\alpha\beta\delta}^2 + n\sigma_{\alpha\beta}^2 + nb\sigma_{\alpha\delta}^2 + nbcc_{\alpha}^2 \\ &= E(MS_{denom}) + nbcc_{\alpha}^2 \end{aligned}$$

Unfortunately, when H_0 is true, F does not have an F-distribution, although an F-distribution with specially computed degrees of freedom provides a pretty good approximation.

Here is an analysis of the data used but not listed in Oehlert Example 11.2. It is artificial data purporting to be measurements of carton strength.

```
Cmd> carton3 <- read("","carton3")
      carton3      400      4
      Read from file "TPl:Stat5303:Data:carton.dat"

Cmd> makecols(carton3,mach,oper,gbat,y)

Cmd> mach <- factor(mach);oper <- factor(oper)
Cmd> gbat <- factor(gbat) # glue batch

Cmd> anova("y=mach*oper*gbat",pval:T)
Model used is y=mach*oper*gbat

```

	DF	SS	MS	P-value
CONSTANT	1	8.6671e+06	8.6671e+06	0
mach	9	2705.8	300.64	3.4897e-16
oper	9	8886.8	987.42	8.0281e-42
mach.oper	81	1682.5	20.772	0.71494
gbat	1	2375.8	2375.8	1.1082e-19
mach.gbat	9	420.48	46.72	0.039738
oper.gbat	9	145.34	16.149	0.71282
mach.oper.gbat	81	1649.8	20.368	0.74902
ERROR1	200	4645.8	23.229	

```
Cmd> ems("y=mach*oper*gbat",vector("mach","oper","gbat"))
Compacting memory, please stand by in macro colproduct
EMS(CONSTANT) = V(ERROR1) + 2V(mach.oper.gbat) + 20V(oper.gbat)
+ 20V(mach.gbat) + 200V(gbat) + 4V(mach.oper) + 40V(oper) +
40V(mach) + 400Q(CONSTANT)
EMS(mach) = V(ERROR1) + 2V(mach.oper.gbat) + 20V(mach.gbat) +
4V(mach.oper) + 40V(mach)
EMS(oper) = V(ERROR1) + 2V(mach.oper.gbat) + 20V(oper.gbat) +
4V(mach.oper) + 40V(oper)
EMS(mach.oper) = V(ERROR1) + 2V(mach.oper.gbat) + 4V(mach.oper)
EMS(gbat) = V(ERROR1) + 2V(mach.oper.gbat) + 20V(oper.gbat) +
20V(mach.gbat) + 200V(gbat)
EMS(mach.gbat) = V(ERROR1) + 2V(mach.oper.gbat) + 20V(mach.gbat)
EMS(oper.gbat) = V(ERROR1) + 2V(mach.oper.gbat) + 20V(oper.gbat)
EMS(mach.oper.gbat) = V(ERROR1) + 2V(mach.oper.gbat) +
EMS(ERROR1) = V(ERROR1)
```

As you can see, my Mac complained about the need for lots of memory to compute the EMS table.

You can check the coefficients match the formulas. For instance $n = 2$ is always the multiplier for $v(\text{mach.oper.gbat}) = \sigma_{\alpha\beta\gamma}^2$ and $nac = 40$ is the multiplier for $v(\text{oper}) = \sigma_{\beta}^2$

Compute the F-statistics to test

$$H_0: \sigma_{\alpha}^2 = 0$$

```
Cmd> ms_num <- MS[2] + MS[8]
Cmd> ms_denom <- MS[4] + MS[6]
Cmd> f_stat <- ms_num/ms_denom; f_stat
(1) 4.7563
```

Q. Since F doesn't really have the F-distribution, how do you use it to test H_0 ?

A. You still use the F-distribution, but with special calculations for degrees of freedom, as an approximation to the distribution when H_0 is true

In this case, the formulas for the degrees of freedom are.

$$df_{num} = \frac{(MS_A + MS_{ABC})^2}{MS_A^2/df_A + MS_{ABC}^2/df_{ABC}}$$

$$= MS_{num}^2 / \{MS_A^2/df_A + MS_{ABC}^2/df_{ABC}\}$$

$$df_{denom} = \frac{(MS_{AB} + MS_{AC})^2}{MS_{AB}^2/df_{AB} + MS_{AC}^2/df_{AC}}$$

$$= MS_{denom}^2 / \{MS_{AB}^2/df_{AB} + MS_{AC}^2/df_{AC}\}$$

This approximation is due to Satterthwaite.

```

Cmd> ms_num <- MS[2] + MS[8]
Cmd> ms_denom <- MS[4] + MS[6]
Cmd> F_stat <- ms_num/ms_denom; F_stat
(1) 4.7563
Cmd> df_num <- ms_num^2/(MS[2]^2/DF[2] + MS[4]^2/DF[4])
Cmd> df_denom <- ms_denom^2/(MS[4]^2/DF[4] + MS[6]^2/DF[6])
Cmd> vector(df_num,df_denom)
(1) 10.255 18.378
Cmd> 1 - cumF(F_stat,df_num,df_denom)
(1) 0.0018512
    
```

Macro `mixed()` does this for you automatically:

```

Cmd> mixed("y=mach*oper*gbat", vector("mach", "oper", "gbat"))
          DF      MS      Error DF      Error MS      F      P value
CONSTANT 1 8.667e+06 2.355 3684 67.49 2353 0.001374
mach      10.26      321 18.38 36.92 4756 1.765e-14
oper      9.375      81 39.74 20.37 27.3 1.02e-05
mach.oper 81 20.77 81 62.87 1.02 0.4648
gbat      1.017      9 46.72 20.37 2.294 0.02386
mach.gbat 9 16.15 81 20.37 0.7929 0.6237
oper.gbat 81 20.37 200 23.23 0.8768 0.749
mach.oper.gbat 200 23.23 0 MISSING MISSING
ERROR1     0
    
```

Here's the **general formula for DF**.

When $MS = \sum_k g_k MS_k$, where MS_k has df_k degrees of freedom, approximately

$$DF = MS^2 / (\sum_k g_k^2 MS_k^2 / df_k)$$

When all the $g_k = 1$, $DF = MS^2 / (\sum_k MS_k^2 / df_k)$

Estimates of variance components

There are several ways to estimate variance components.

Simplest and easiest to understand:

Use a **linear combination of MS** that has the proper expectation.

For the one-way balanced case

$$EMS_A = \sigma^2 + n\sigma_{\alpha}^2 \text{ and } EMS_{error} = \sigma^2$$

so $(EMS_A - EMS_{error})/n = \sigma_{\alpha}^2$ and

$$\hat{\sigma}_{\alpha}^2 = (MS_A - MS_{error})/n \text{ is unbiased}$$

For the two-way balanced case:

$$EMS_A = \sigma^2 + n\sigma_{\alpha\beta}^2 + nb\sigma_{\alpha}^2$$

$$EMS_{AB} = \sigma^2 + n\sigma_{\alpha\beta}^2, EMS_{error} = \sigma^2$$

Then $(EMS_{AB} - EMS_{error})/n = \sigma_{\alpha\beta}^2$

$$(EMS_A - EMS_{AB})/(nb) = \sigma_{\alpha}^2$$

So unbiased estimates are

$$\hat{\sigma}_{\alpha\beta}^2 = (MS_{AB} - MS_{ABC})/n$$

$$\hat{\sigma}_{\alpha}^2 = (MS_A - MS_{AB})/(nb)$$

For the three-way balanced case, since $EMS_A - EMS_{AB} - EMS_{AC} + EMS_{ABC} = nb\sigma_{\alpha}^2$

$$\hat{\sigma}_{\alpha}^2 = (MS_A - MS_{AB} - MS_{AC} + MS_{ABC})/nbc$$

is unbiased.

```
Cmd> (MS[2]-MS[4]-MS[6]+MS[8])/(2*2*10)
(1) 6.338
```

You can calculate approximate degrees of freedom similarly as before as $df =$

$$(MS_A - MS_{AB} - MS_{AC} - MS_{ABC})^2$$

$$MS_A^2/df_A + MS_{AB}^2/df_{AB} + MS_{AC}^2/df_{AC} + MS_{ABC}^2/df_{ABC}$$

```
Cmd> J <- vector(2,4,6,8)
Cmd> (MS[2]-MS[4]-MS[6]+MS[8])^2/sum(MS[J]^2/DF[J])
(1) 6.2425
```

`varcomp()` does black box computations.

```
Cmd> varcomp("y=mach*oper*gbat", vector("mach", "oper", "gbat"))
      Estimate      SE      DF
mach      6.338      3.5875      6.2425
oper      24.272      11.639      8.6976
mach.oper  0.10114      1.1428      0.015664
gbat      11.666      16.8      0.96449
mach.gbat  1.3176      1.1128      2.8042
oper.gbat  -0.21093      0.41291      0.52191
mach.oper.gbat -1.4307      1.9773      1.0471
ERROR1    23.229      2.3229      200
```

SE is almost meaningless here because sample sizes are very small.