Displays for Statistics 5303

Lecture 29

November 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

Inference on variance components

All the methods methods for variance component inverence depend strongly on normality of

- the errors ε_{ijk}
- the random effects α_i, β_j, $\alpha\beta_{ij}$, ...

The following "facts" assume all the random variables involved are normal and independent and σ^2 is constant

In the balanced case each MS is distributed as

$$EMS\times\chi_{df}^2/df$$

This is the basis for the standard confidence interval for EMS:

$$Conf(MS/\{\chi_{\varepsilon/2}^2/df\} \leq EMS \leq MS/\{\chi_{1-\varepsilon/2}^2/df\}) = 1 - \varepsilon$$

Note: An *upper* χ^2 probability point is in the denominator of the *lower* limit while a *lower* χ^2 probability point is in the denominator of the *upper* limit.

The more usual way to write these limits is

$$df\times MS/\chi_{\varepsilon/2}^2 \leq EMS \leq df\times MS/\chi_{1-\varepsilon/2}^2$$

but I prefer to put χ^2/df in the denominator.

In particular, you can use these limits for $\sigma^2 = EMS_{error}$

$$df_{error}\times MS_E/\chi_{\varepsilon/2}^2 \leq \sigma^2 \leq df\times MS_E/\chi_{1-\varepsilon/2}^2$$

where the χ^2 degrees of freedom = df_{errror}.

It's sometimes helpful to know the mean and variance of χ_{df}^2, χ_{df}^2/df and F_{df_1,df_2}.

	Mean	Variance
χ_{df}^2	df	1
χ_{df}^2/df	2df	2/df
F_{df_1,df_2}	$df_2/(df_2-2)$	(more complicated)

I'm using Oehlert's Carton Experiment 3 data as an example:

```
Cmd> carton3 <- read("","carton3")
carton3        400      4
) Artificial data used Example 11.2, Oehlert p. 263
) There are two replicates of 10^2x2 factorial data from
) an imagined experiment studying the variability in carton
) strength due to variability among machines, operators and
) glue batch
) Col. 1: Machine (1 - 10)
) Col. 2: Operator (1 - 10)
) Col. 3: Glue batch (1 - 2)
) Col. 4: Strength of carton
Read from file "TP1:Stat5303:Data:carton.dat"

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

Cmd> mach <- factor(mach);oper <- factor(oper)

Cmd> gluebat <- factor(gluebat)

Cmd> a <- b <- 10; c <- 2; n <- 2

Cmd> anova("y=mach*oper*gluebat",silent:T)

Cmd> MS <- SS/DF

Cmd> MS # The ANOVA mean squares
    CONSTANT        mach        oper   mach.oper     gluebat
mach.gluebatoper.gluebatmach.oper.gluebat       ERROR1
  8.6671e+06      300.64      987.42      20.772      2375.8
       46.72      16.149      20.368      23.229

Cmd> DF# The ANOVA degrees of freedom
    CONSTANT        mach        oper   mach.oper     gluebat
mach.gluebatoper.gluebatmach.oper.gluebat       ERROR1
           1           9           9          81           1
           9           9          81         200
```

With real data, before proceeding you should look at residual plots, see if you need a transformation, and so on.

Since $EMS_{error} = \sigma^2$, the "exact" χ^2 applies.

```
Cmd> sigmasqhat <- reverse(MS)[1]; sigmasqhat
(1)      23.229
```

`reverse(MS)[1]` is a "trick" to get the last MS, the error MS. In this case it's the same as `MS[9]`.

```
Cmd> eps <- .10 # for 90% confidence
Cmd> sigmasqhat/(invchi(vector(1-eps/2,eps/2),200)/200)
(1)      19.854      27.608
```

You can use this same method when you want limits for $EMS_{ABC} = \sigma^2 + n\sigma_{\alpha\beta\gamma}$ or any other EMS.

It's also fairly easy to get "exact" limits for EMS_1/EMS_2 (say $EMS_{ABC}/EMS_{error} = 1+n\sigma_{\alpha\beta\gamma}^2/\sigma^2$) because

$$F = MS_1/MS_2 = (EMS_1/EMS_2)F_{df_1,df_2}$$

Lower limit $= (MS_1/MS_2)/F_{\epsilon/2,df_1,df_2}$

$= F_{observed}/F_{\epsilon/2,df_1,df_2}$

Upper limit $= F_{observed}/F_{1-\epsilon/2,df_1,df_2}$

Example

```
Cmd> f_abc <- MS[8]/MS[9] # MS_ABC/MS_error
Cmd> limits <- f_abc/invF(vector(1-eps/2,eps/2),DF[8],DF[9])
Cmd> limits
(1)     0.65203     1.2068
```

These are limits for

$$(\sigma^2+n\sigma_{\alpha\beta\gamma}^2)/\sigma^2 = 1 + n\sigma_{\alpha\beta\gamma}^2/\sigma^2$$

From these you can get exact limits for the ratio $\sigma_{\alpha\beta\gamma}^2/\sigma^2 = ((\sigma^2+n\sigma_{\alpha\beta\gamma}^2)/\sigma^2 - 1)/n$.

```
Cmd> (limits - 1)/n
(1)    -0.17398     0.1034
```


Exact vs approximate inference

So far the inference methods for variance components have been "exact".

That is, when all the conditions are satisfied, confidence intervals have exactly the intended confidence level $1 - \epsilon$ and tests have exactly the intended type I error probability ϵ.

But many confidence intervals for variance components are only approximate in the sense that the actual confidence level is not exactly the intended level $1 - \epsilon$.

There are at least two approaches to approximate inference for variance components.

Chi-squared approximation

Even when it is not exactly correct, treat an estimated variance component $\hat{\sigma}_X^2$ *as if* it were distributed as $\sigma_X^2\chi_{df}^2/df$, where X is any label such as α, β, $\alpha\gamma$, ...

Then, if you can provide a value for df, approximate confidence limits are

$$\hat{\sigma}_X^2/(\chi_{\epsilon/2,df}^2/df) \leq \sigma_X^2 \leq \hat{\sigma}_X^2/(\chi_{1-\epsilon/2,df}^2/df)$$

Degrees of freedom

When $\hat{\sigma}_X^2 = \sum_j g_j MS_j$,

$$\hat{df} = \hat{\sigma}_X^4/(\sum_j g_j^2 MS_j^2/df_j)$$

This is based on a formula for $V[\hat{\sigma}_X^2]$ which is correct *only* **when data are normal.**

Example:

Estimate of σ_α^2 in 3-factor random effects ANOVA.

The estimate is

$$(MS_A - MA_{AB} - MS_{AC} + MA_{ABC})/(bcn)$$

for which $g_A = -g_{AB} = -g_{AC} = g_{ABC} = 1/bcn$

```
Cmd> sig_Asq_hat <- (MS[2] - MS[4] - MS[6] + MS[8])/(b*c*n)

Cmd> sig_Asq_hat #estimated variance component
(1)       6.338

Cmd> g <- vector(1,-1,-1,1)/(b*c*n) # g-coefficients

Cmd> J <- vector(2,4,6,8)

Cmd> sum(g*MS[J]) # another way
(1)       6.338   sig_Asq_hat computed another way

Cmd> df_Asq_hat <- sig_Asq_hat^2/sum(g^2*MS[J]^2/DF[J])

Cmd> df_Asq_hat
(1)       6.2425   df for sig_Asq_hat
```

Use these to get a confidence interval for σ_α^2.

```
Cmd> sig_Asq_hat/\
     (invchi(vector(1-eps/2,eps/2),df_Asq_hat)/df_Asq_hat)
(1)      3.0545      22.469
```


This is the "white box" way. MacAnova macro **varcomp()** provides a "black box" way.

One way to use **varcomp()** is to first compute and save the information on EMSs using keyword phrase **keep:T** on **ems()**.

```
Cmd> EMS <- ems("y=mach*oper*gluebat",\
     vector("mach","oper","gluebat"),keep:T)
Compacting memory, please stand by in macro ems
```

The result, **EMS**, is a structure that **varcomp()** knows how to use.

```
Cmd> compnames(EMS)
(1) "df"           Usual ANOVA DF
(2) "ss"           Usual ANOVA SS
(3) "termnames"    Usual ANOVA term names
(4) "coefs"        Multipliers in EMS formulas
(5) "rterms"       Vector of T's and F's; T => random term

Cmd> print(format:"3.0f",EMS$coefs)
SCRATCH:
(1,1) 400  40  40   4 200  20  20   2   1
(2,1)   0  40   0   4   0  20   0   2   1
(3,1)   0   0  40   4   0   0  20   2   1
(4,1)   0   0   0   4   0   0   0   2   1
(5,1)   0   0   0   0 200  20  20   2   1
(6,1)   0   0   0   0   0  20   0   2   1
(7,1)   0   0   0   0   0   0  20   2   1
(8,1)   0   0   0   0   0   0   0   2   1
(9,1)   0   0   0   0   0   0   0   0   1
```


varcomp(EMS) gives the same output as

**varcomp("y=mach*oper*gluebat",\
vector(mach","oper","gluebat"))**

```
Cmd> varcomp(EMS)
WARNING: searching for unrecognized macro varcomp near
varcomp(
                      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
gluebat                 11.666       16.8     0.96449
mach.gluebat            1.3176     1.1128      2.8042
oper.gluebat          -0.21093    0.41291     0.52191
mach.oper.gluebat      -1.4307     1.9773      1.0471
ERROR1                  23.229     2.3229         200
```

Note the estimate and DF for machines match the white box values. You can use the values directly from the table to get a C.I. for $\sigma_\beta^2 = \sigma_{oper}^2$:

```
Cmd> 24.272/(invchi(vector(1-eps/2,eps/2),8.6976)/8.6976)
(1)      12.798      67.16
```


Normal approximation

For large degrees of freedom,

$$\sigma_X^2\chi_{df}^2/df \cong N(\sigma_X^2, 2\sigma_X^4/df)$$

This suggests a confidence interval based on the normal distribution

$$\sigma_X^2 = \hat\sigma_X^2 \pm z_{\alpha/2}\sqrt{\{2\hat\sigma_X^4/\hat{df}_X\}}$$
$$= \hat\sigma_X^2 \pm z_{\alpha/2}\sqrt{\{2\textstyle\sum_j g_j^2 MS_j^2/df_j\}}$$

because

$$\hat{df}_X = \hat\sigma_X^4/(\textstyle\sum_j g_j^2 MS_j^2/df_j)$$

The values in the **SE** column of the **varcomp()** output are $\sqrt{\{2\sum_j g_j^2 MS_j^2/df_j\}}$

```
Cmd> se_comp <- sqrt(2*sum(g^2*MS[J]^2/DF[J])); se_comp
(1)       3.5875      Same as in output

Cmd> sum(g*MS[J]) + 1.96*vector(-1,1)*se_comp
(1)     -0.69346      13.369
```

This last would be an approximate 95% interval for σ_α^2 if the degrees of freedom were quite large, which they are not.

You can do better by finding limits for σ_x based on a normal approximation to $\sqrt{\{\chi^2\}}$ whose distribution is much *less skewed* than the distribution of χ^2.

The approximate variance of $\hat{\sigma}_x$ is

$$\mathrm{Var}(\hat{\sigma}_x{}^2) = \sigma_x{}^2/(2\times \mathrm{df}).$$

Here are approximate limits for σ_α.

```
Cmd> sqrt(sig_Asq_hat) + \
      invnor(1-eps/2)*vector(-1,1)*sqrt(sig_Asq_hat)/df_Asq_hat
(1)      1.8542      3.1809
```

Square these to get approximate limits for $\sigma_\alpha{}^2$:

```
Cmd> (sqrt(sig_Asq_hat) + \
 invnor(1-eps/2)*vector(-1,1)*sqrt(sig_Asq_hat)/df_Asq_hat)^2
(1)       3.438      10.118
```

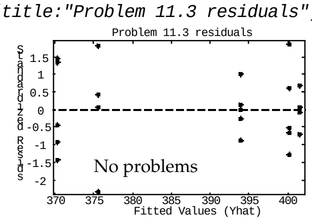
Compare these with the values computed directly from χ^2:

```
Cmd> sig_Asq_hat/\
     (invchi(vector(1-eps/2,eps/2),df_Asq_hat)/df_Asq_hat)
(1)      3.0545      22.469
```

The lower limit is not bad, but the upper limit is far off.

Here is an analysis of the data in Oehlert Problem 11.3, a one-factor problem:

```
Cmd> data <- read("","pr11.3")
pr11.3          25     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 Vangel, M.~G. (1992). ``New methods for
) one-sidedtolerance limits for a one-way balanced random-
) effects {ANOVA} model.''{\em Technometrics\/}~{\em 34},
) 176--185.
)
) Problem 11.3, p. 278
) Columns are batch number and (coded) tensile strength.
Read from file "TP1:Stat5303:Data:OeCh11.dat"

Cmd> makecols(data,batno,tensile)

Cmd> batno <- factor(batno)

Cmd> anova("tensile=batno")
Model used is tensile=batno
                DF          SS          MS
CONSTANT         1  3.7706e+06  3.7706e+06
batno            4      4163.4      1040.8
ERROR1          20      1578.4       78.92

Cmd> tabs(tensile,batno,count:T) # it's balanced
(1)          5           5           5           5           5

Cmd> n <- 5; a <- 5

Cmd> resvsyhat(title:"Problem 11.3 residuals")
```




```
Cmd>  ems("tensile=batno","batno")
EMS(CONSTANT) = V(ERROR1) + 5V(batno) + 25Q(CONSTANT)
EMS(batno) = V(ERROR1) + 5V(batno)
EMS(ERROR1) = V(ERROR1)

Cmd> vcomp <- varcomp("tensile=batno","batno"); vcomp
         Estimate          SE          DF
batno      192.38      147.28      3.4125
ERROR1      78.92      24.957          20

Cmd> MS <- SS/DF

Cmd> MS
    CONSTANT       batno      ERROR1
  3.7706e+06      1040.8       78.92

Cmd> (MS[2] - MS[3])/n # estimate of sigma_alpha^2
(1)      192.38
```

This is an estimate of $\sigma_\alpha{}^2$.

Now find a 90% confidence interval.

```
Cmd> eps <- .10 # (1 - .9)

Cmd> estimate <- vcomp[1,1]; estimate # from varcomp() output
        Estimate
batno     192.38

Cmd> df <- vcomp[1,3]; df # from varcomp() output
              DF
batno     3.4125

Cmd> vector(df*estimate/invchi(vector(1-eps/2,eps/2),df))
(1)       77.069      1342.9    90% confidence interval
```


Now get limits for $\sigma_\alpha{}^2/\sigma^2$:

```
Cmd> f <- MS[2]/MS[3] # F-statistic

Cmd> limits <- f/invF(vector(1-eps/2,eps/2),DF[2],DF[3])

Cmd> limits
(1)      4.6016      76.527
```

These are 90% limits for

$$(\sigma^2 + n\sigma_\alpha{}^2)/\sigma^2 = 1 + n\sigma_\alpha{}^2/\sigma^2$$

```
Cmd> (limits - 1)/n
(1)     0.72032      15.105
```

These are limits for $\sigma_\alpha{}^2/\sigma^2$

Power of F-tests of H_0: $\sigma_X^2 = 0$

In balanced case, ratios of mean squares in the ANOVA tables have the distribution of a multiple of F:

$$F = MS_1/MS_2 = (EMS_1/EMS_2)F_{df_1,df_2}$$

The power of a test of

H_0: $EMS_1 = EMS_2$, that is H_0: $EMS_1/EMS_2 = 1$, against the specific alternative

$$H_a: EMS_1/EMS_2 = \rho \neq 1$$

is

$$\begin{aligned} \text{Power} &= P(MS_1/MS_2 > F_{\varepsilon,df_1,df_2}) \\ &= P(\rho F_{df_1,df_2} > F_{\varepsilon,df_1,df_2}) \\ &= P(F_{df_1,df_2} > (1/\rho)F_{\varepsilon,df_1,df_2}) \end{aligned}$$

You can use this when

$$EMS_1 = EMS_2 + K\times\sigma_X^2$$

For example, when $EMS_1 = \sigma^2 + n\sigma_\alpha^2$ and $EMS_2 = \sigma^2$ so $\rho = 1 + n\sigma_\alpha^2/\sigma^2$.

In this example, suppose $\sigma^2 = 80$ and $\sigma_\alpha^2 = 200$ (close to estimates). Let's find the power of a 5% test of H_0: $\sigma_\alpha^2 = 0$ for a range of values of n.

```
Cmd> sigmasq <- 80; sigma_Asq <- 200

Cmd> n <- run(2,30); a <- 5

Cmd> critvals <- invF(1-.05,a-1,a*(n-1)); critvals
 (1)     5.1922       3.478      3.0556      2.8661      2.7587
 (6)     2.6896      2.6415       2.606      2.5787      2.5572
(11)     2.5397      2.5252       2.513      2.5027      2.4937
(16)     2.4859       2.479      2.4729      2.4675      2.4626
(21)     2.4582      2.4542      2.4506      2.4472      2.4442
(26)     2.4414      2.4387      2.4363       2.434

Cmd> ems2 <- sigmasq; ems1 <- sigmasq + n*sigma_Asq

Cmd> rho <- ems1/ems2

Cmd> p <- 1 - cumF((1/rho)*critvals,a-1,a*(n-1)); p
 (1)    0.54293     0.79823     0.88776     0.92853     0.95047
 (6)    0.96364     0.97216       0.978     0.98218     0.98526
(11)    0.98761     0.98944     0.99089     0.99206     0.99302
(16)    0.99382     0.99448     0.99505     0.99553     0.99594
(21)     0.9963     0.99662     0.99689     0.99714     0.99735
(26)    0.99754     0.99772     0.99787     0.99801
```



```
Cmd> lineplot(n,p,title:"Power of 5% test vs n",\
     ylab:"Power",ymin:0)
```