Statistics 5401
16. Testing Multivariate Linear Hypotheses

Gary W. Oehlert
School of Statistics
313B Ford Hall
612-625-1557
gary@stat.umn.edu

In the univariate linear model, we test the null hypothesis H$_0$: $\mathbf{L}\beta = 0$ by using the error sum of squares SS_E and the hypothesis sum of squares SS_H, with f_e and f_h degrees of freedom respectively.

```
Cmd> readdata("",ext,add,x1,x2,x3)
Read from file "~/JW5data/T6-4.dat"
Column 1 saved as REAL vector ext
Column 2 saved as REAL vector add
Column 3 saved as REAL vector x1
Column 4 saved as REAL vector x2
Column 5 saved as REAL vector x3

Cmd> a <- makefactor(ext)

Cmd> b <- makefactor(add)

Cmd> anova("x1=a*b",fstats:T)
Model used is x1=a*b
                 DF          SS          MS           F      P-value
CONSTANT          1      920.72      920.72  8351.24263            0
a                 1      1.7405      1.7405    15.78685    0.0010917
b                 1      0.7605      0.7605     6.89796      0.01833
a.b               1      0.0005      0.0005     0.00454      0.94714
ERROR1           16       1.764     0.11025
```

The SS_H for testing no factor A effect is 1.74 (1 df), the SS_E is 1.764 (16 df).
Under H$_0$:

$$SS_E \sim \sigma^2\chi^2_{f_e} \qquad\qquad SS_H \sim \sigma^2\chi^2_{f_h}$$

SS_H will tend to be larger than $\sigma^2\chi^2_{f_h}$ when H$_0$ is false, so we test via

$$\frac{f_e}{f_h}\frac{SS_H}{SS_E} \sim F_{f_h,\,f_e}$$

or

$$f_e\frac{SS_H}{SS_E} \sim f_h F_{f_h,\,f_e}$$

For large f_e,

$$f_e\frac{SS_H}{SS_E} \sim \chi^2_{f_h}$$

The F-based p-value can be read off the output. The large-sample approximate χ^2 is

```
Cmd> 16*1.7405/1.765
(1)       15.778

Cmd> 1-cumchi(15.778,1)
(1)   7.1226e-05
```

The small sample F p-value (.0011) and large sample χ^2 p-value (.00007) are not very close, but $f_e = 16 \neq \infty$. For a multivariate linear model, when we want to test a null hypothesis $H_0 : \mathbf{L}\mathcal{B} = 0$, we get hypothesis and error matrices $\mathbf{H}$ and $\mathbf{E}$.

```
Cmd> Y <- hconcat(x1,x2,x3)

Cmd> manova("Y=a*b",print:F)
Model used is Y=a*b

Cmd> h <- SS[2,,];h
               (1)          (2)          (3)
a (1)       1.7405      -1.5045       0.8555
  (2)      -1.5045       1.3005      -0.7395
  (3)       0.8555      -0.7395       0.4205

Cmd> e <- SS[5,,];e
                (1)      (2)      (3)
ERROR1 (1)    1.764     0.02    -3.07
       (2)     0.02    2.628   -0.552
       (3)    -3.07   -0.552   64.924

Cmd> coefs("a")
(1,1)   -0.295    0.255   -0.145
(2,1)    0.295   -0.255    0.145
```

The real $\hat{\mathcal{B}}$ does not contain the redundant coefficient found by the constraint that the coefficients sum to zero. Here is $\hat{\mathcal{B}}$:

```
Cmd> modelinfo(coefs:T)
(1,1)    6.785    9.315    3.935
(2,1)   -0.295    0.255   -0.145
(3,1)   -0.195   -0.175   -0.495
(4,1)    0.005    0.165    0.445
```

$\mathbf{X}$

```
Cmd> modelinfo(xvars:T)
 (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
(17,1)	1	-1	-1	1
(18,1)	1	-1	-1	1
(19,1)	1	-1	-1	1
(20,1)	1	-1	-1	1

Suppose now that we wish to test that factor A has no effect is the multivariate model. The only testing method we know now is Hotelling's T^2. Let's set things up for T^2.

We want to test whether the second row of $\mathcal{B}$ is zero. If we unroll $\mathcal{B}$ into a vector, that is testing that the second, sixth, and tenth elements of $\mathbf{b}$ are zero.

Let $\mathbf{L} = [0, 1, 0, 0]$. Then we are trying to test

$$H_0 : (\mathbf{I}_3 \otimes \mathbf{L})\ \mathbf{b} = \begin{bmatrix} \mathbf{b}_2 \\ \mathbf{b}_6 \\ \mathbf{b}_{10} \end{bmatrix} = 0$$

Recall that the distribution of $\hat{\mathbf{b}}$ is

$$\hat{\mathbf{b}} \sim \mathrm{N}_{rp}(\mathbf{b}, \Sigma \otimes (\mathbf{X}'\mathbf{X})^{-1})$$

so that

$$\begin{aligned} (\mathbf{I}_p \otimes \mathbf{L})\ \hat{\mathbf{b}} &\sim \mathrm{N}_p((\mathbf{I}_p \otimes \mathbf{L})\mathbf{b}, \mathbf{I}_p \Sigma \mathbf{I}_p \otimes \mathbf{L}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{L}') \\ &= \mathrm{N}_p(\begin{bmatrix} \mathbf{b}_2 \\ \mathbf{b}_6 \\ \mathbf{b}_{10} \end{bmatrix}, \Sigma \times (\mathbf{X}'\mathbf{X})^{-1}_{22}) \end{aligned}$$

$$\begin{aligned} T^2 &= [\hat{\mathbf{b}}_2, \hat{\mathbf{b}}_6, \hat{\mathbf{b}}_{10}] \frac{\mathbf{S}^{-1}}{(\mathbf{X}'\mathbf{X})^{-1}_{22}} \begin{bmatrix} \hat{\mathbf{b}}_2 \\ \hat{\mathbf{b}}_6 \\ \hat{\mathbf{b}}_{10} \end{bmatrix} \\ &= \text{trace}\left([\hat{\mathbf{b}}_2, \hat{\mathbf{b}}_6, \hat{\mathbf{b}}_{10}] \frac{\mathbf{S}^{-1}}{(\mathbf{X}'\mathbf{X})^{-1}_{22}} \begin{bmatrix} \hat{\mathbf{b}}_2 \\ \hat{\mathbf{b}}_6 \\ \hat{\mathbf{b}}_{10} \end{bmatrix} \right) \\ &= f_e\ \text{trace}\left(\frac{\mathbf{E}^{-1}}{(\mathbf{X}'\mathbf{X})^{-1}_{22}} \begin{bmatrix} \hat{\mathbf{b}}_2 \\ \hat{\mathbf{b}}_6 \\ \hat{\mathbf{b}}_{10} \end{bmatrix} [\hat{\mathbf{b}}_2, \hat{\mathbf{b}}_6, \hat{\mathbf{b}}_{10}] \right) \\ &= f_e\ \text{trace}\left(\mathbf{E}^{-1}\mathbf{H} \right) \end{aligned}$$

Using the trace formulation.

```
Cmd> trace(16*solve(e)%*%h)
(1)         25.9
```

Now redo using Kronecker products, etc.

```
Cmd> L <- vector(0,1,0,0)'

Cmd> I <- dmat(rep(1,3))

Cmd> kronecker <- macro("
@A <- matrix($1)
@B <- matrix($2)
@C <- outer(@A,@B)
@C <- t(@C,vector(3,1,4,2))
@C <- matrix(@C,nrows(@A)*nrows(@B))
@C")

Cmd> print(kronecker(I,L),format:"f2.0")
MATRIX:
(1,1)  0  1  0  0  0  0  0  0  0  0  0  0
(2,1)  0  0  0  0  0  1  0  0  0  0  0  0
(3,1)  0  0  0  0  0  0  0  0  0  1  0  0

Cmd> IkL <- kronecker(I,L)

Cmd> bhat <- vector(modelinfo(coefs:T))

Cmd> IkL%*%bhat
(1,1)       -0.295
(2,1)        0.255
(3,1)       -0.145

Cmd> modelinfo(coefs:T)
(1,1)        6.785        9.315        3.935
(2,1)       -0.295        0.255       -0.145
(3,1)       -0.195       -0.175       -0.495
(4,1)        0.005        0.165        0.445

Cmd> v <- kronecker(e/16,L%*%xtxinv%*%L')

Cmd> IkLb <- IkL%*%bhat

Cmd> IkLb'%*%solve(v)%*%IkLb
(1,1)         25.9
```

Now

$$T^2 \sim \frac{pf_e}{f_e - (p-1)} F_{p,\, f_e-(p-1)}$$

so

$$\frac{f_e - (p-1)}{p f_e} T^2 \sim F_{p,\, f_e-(p-1)}$$

or

$$\frac{f_e - (p-1)}{f_e} T^2 \sim p\, F_{p,\, f_e-(p-1)}$$

or for large f_e

$$\frac{f_e - (p-1)}{f_e} T^2 \sim \chi^2_p$$

Small sample, F-based p-value.

```
Cmd> (16-2)/3/16*25.9
(1)       7.5542

Cmd> 1-cumF(7.5542,3,14)
(1)    0.0030342
```

Large sample version with χ^2 approximation:

```
Cmd> (16-2)/16*25.9
(1)       22.662

Cmd> 1-cumchi(22.662,3)
(1)   4.7493e-05
```

Again, with $f_e = 16$, the large sample approximate p-value is not all that close to the small sample p-value.
Now let's try the same thing with a multi-row $\mathbf{L}$, that is, an $\mathbf{L}$ that corresponds to a more than one degree of freedom test in each variable.
Let $\mathbf{L}$ be $q \times r$ with $1 < q \leq r$ with rank q.

$$(\mathbf{I}_p \otimes \mathbf{L})\ \hat{\mathbf{b}} \sim \mathrm{N}_{pq}((\mathbf{I}_p \otimes \mathbf{L})\mathbf{b}, \mathbf{I}_p \Sigma \mathbf{I}_p \otimes \mathbf{L}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{L}')$$

We can form

$$\begin{aligned} T^2 &= ((\mathbf{I}_p \otimes \mathbf{L})\ \hat{\mathbf{b}})'(\mathbf{S} \otimes \mathbf{L}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{L}')^{-1}(\mathbf{I}_p \otimes \mathbf{L})\ \hat{\mathbf{b}} \\ &= f_e\ \mathrm{tr}\left((\mathbf{I}_p \otimes \mathbf{L})'(\mathbf{E} \otimes \mathbf{L}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{L}')^{-1}(\mathbf{I}_p \otimes \mathbf{L})\ \hat{\mathbf{b}}\hat{\mathbf{b}}'\right) \\ &= f_e\ \mathrm{tr}\left((\mathbf{I}_p \otimes \mathbf{L})'(\mathbf{E}^{-1} \otimes (\mathbf{L}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{L}')^{-1})(\mathbf{I}_p \otimes \mathbf{L})\ \hat{\mathbf{b}}\hat{\mathbf{b}}'\right) \\ &= f_e\ \mathrm{tr}\left((\mathbf{E}^{-1} \otimes \mathbf{L}'(\mathbf{L}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{L}')^{-1}\mathbf{L})\hat{\mathbf{b}}\hat{\mathbf{b}}'\right) \\ &= f_e\ \mathrm{tr}(\mathbf{E}^{-1}\mathbf{H}) \end{aligned}$$

8401 guys should prove that last step.

```
Cmd> readdata("",x1,x2,x3,x4,a)
Read from file "~/JW5data/T6-13.DAT"
Column 1 saved as REAL vector x1
```

```
Column 2 saved as REAL vector x2
Column 3 saved as REAL vector x3
Column 4 saved as REAL vector x4
Column 5 saved as REAL vector a

Cmd> Y <- hconcat(x1,x2,x3,x4)

Cmd> a <- factor(a)

Cmd> manova("Y=a")
Model used is Y=a
...

Cmd> h <- matrix(SS[2,,])

Cmd> e <- matrix(SS[3,,])

Cmd> 87*trace(solve(e)%*%h)
(1)       17.564

Cmd> B <- modelinfo(coefs:T);B
(1,1)    132.73    133.37    98.089    50.444
(2,1)   -1.3667   0.23333    1.0778  0.088889
(3,1)  -0.36667  -0.66667   0.97778  -0.21111

Cmd> L <- matrix(vector(0,0,1,0,0,1),2);L
(1,1)            0            1            0
(2,1)            0            0            1

Cmd> L %*% B
(1,1)   -1.3667   0.23333    1.0778  0.088889
(2,1)  -0.36667  -0.66667   0.97778  -0.21111

Cmd> b <- vector(B)

Cmd> I4 <- dmat(rep(1,4))

Cmd> IkL <- kronecker(I4,L)

Cmd> IkL%*%b
(1,1)      -1.3667
(2,1)     -0.36667
(3,1)      0.23333
(4,1)     -0.66667
(5,1)       1.0778
```

```
(6,1)      0.97778
(7,1)     0.088889
(8,1)     -0.21111

Cmd> xtxinv <- modelinfo(xtxinv:T)

Cmd> v <- kronecker(e/87,L%*%xtxinv%*%L')

Cmd> (IkL%*%b)'%*%solve(v)%*%(IkL%*%b)
(1,1)       17.564
```

But ...

$$(\mathbf{S} \otimes \mathbf{L}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{L}')$$

does *not* have a Wishart distribution, so we cannot use the usual F as a reference distribution. `:-(`
The trace of $f_e\mathbf{E}^{-1}\mathbf{H}$ is still a decent test statistic, we just need to find another way to assess p-values.
Note that the trace of $\mathbf{E}^{-1}\mathbf{H}$ is the sum of the eigenvalues of $\mathbf{E}^{-1}\mathbf{H}$.
What about a likelihood ratio test?

$$\begin{aligned}
\Lambda &= \left(\frac{|E|}{|E+H|}\right)^{n/2} \\
&= \left(\frac{1}{|E|^{-1}|E+H|}\right)^{n/2} \\
&= \left(\frac{1}{|E^{-1}||E+H|}\right)^{n/2} \\
&= \left(\frac{1}{|I+E^{-1}H|}\right)^{n/2} \\
&= \left(\prod_{i=1}^{p}(1+\eta_i)\right)^{-n/2}
\end{aligned}$$

where the η_i are the eigenvalues of $\mathbf{E}^{-1}\mathbf{H}$.
Recall that $-2\ln(\Lambda)$ is asymptotically χ^2 with degrees of freedom determined by the number of tested parameters.

$$\begin{aligned}
-2\ln\Lambda &= n\ln\left(\prod_{i=1}^{p}(1+\eta_i)\right) \\
&= n\sum_{i=1}^{p}\ln(1+\eta_i) \\
&\to \chi^2_{pq}
\end{aligned}$$

So the eigenvalues of $\mathbf{E}^{-1}\mathbf{H}$ keep returning.

```
Cmd> det(e)/det(e+h)
(1)       0.8301
```

```
Cmd> length(a)
(1)          90

Cmd> 90*log(det(e+h)/det(e))
(1)      16.759

Cmd> 1-cumchi(16.759,8)
(1)    0.032719
```

Let's look more closely at the magical $\mathbf{E}^{-1}\mathbf{H}$.
Suppose that

$$\mathbf{H}u = \eta\mathbf{E}u$$

Then η is an eigenvalue of $\mathbf{H}$ relative to $\mathbf{E}$ (or a relative eigenvalue), and u is the corresponding eigenvector.
The eigenvalues of $\mathbf{H}$ relative to $\mathbf{E}$ are the eigenvalues of $\mathbf{E}^{-1}\mathbf{H}$.
If u_1 is the relative eigenvector corresponding to the largest relative eigenvalue η_1, then

$$\frac{u'\mathbf{H}u}{u'\mathbf{E}u}$$

is maximized for $u = u_1$. That is, u_1 gives the linear combination of variables that leads to the largest possible "F" statistic for this hypothesis.
Note, however, $u_1'\mathbf{H}u_1/u_1'\mathbf{E}u_1$ does *not* have an F distribution. This is due to the maximization involved in finding u_1.

```
Cmd> releigen(h,e)
component: values
(1)   0.18697  0.014913         0         0
component: vectors
          (1)       (2)       (3)       (4)
(1)  -0.01731  -0.01243 -0.000953  0.011956
(2)  -0.00317   0.01722 -0.011639  0.009657
(3)   0.01600  -0.00746 -0.001343  0.012517
(4)   0.00492   0.01318  0.032188  0.006710

Cmd> vals <- releigen(h,e)$values

Cmd> vecs <- releigen(h,e)$vectors

Cmd> (h %*% vecs[,1])/(e %*% vecs[,1])
             (1)
(1)      0.18697
(2)      0.18697
(3)      0.18697
(4)      0.18697

Cmd> sum(vals)*87
```

```
(1)      17.564

Cmd> 1/prod(1+vals)
(1)      0.8301
```

Further properties of relative eigenvectors.

$$\begin{aligned} u_j'\mathbf{H}u_j &= \eta_j u_j \\ u_j'\mathbf{H}u_k &= 0, \quad j \neq k \\ u_j'\mathbf{E}u_j &= 1 \\ u_j'\mathbf{E}u_k &= 0, \quad j \neq k \end{aligned}$$

u_j is orthogonal to both $\mathbf{H}u_k$ and $\mathbf{E}u_k$ for $k \neq j$.

Multivariate test statistics

Let $\mathbf{H}$ and $\mathbf{E}$ be hypothesis and error matrices with relative eigenvalues η_i.

$$\begin{aligned} \text{Wilks' lambda} &= \prod_{i=1}^{p} \frac{1}{1+\eta_i} = \frac{|\mathbf{E}|}{|\mathbf{E}+\mathbf{H}|} \\ \text{Hotelling-Lawley trace} &= \sum_{i=1}^{p} \eta_i = \mathrm{tr}(\mathbf{E}^{-1}\mathbf{H}) \\ \text{Pillai's trace} &= \sum_{i=1}^{p} \frac{\eta_i}{1+\eta_i} = \mathrm{tr}(\mathbf{H}(\mathbf{E}+\mathbf{H})^{-1}) \\ \text{Roy's greatest root} &= \frac{\eta_1}{1+\eta_1} \end{aligned}$$

We want to reject for large η_i, so Wilks' lambda rejects for small values of the test statistic, whereas the others reject for large values of the test statistic.
The sample size times the Hotelling-Lawley, Pillai's, and log of Wilks' statistics are asymptotically χ^2_{qp} under the null. You need special tables for Roy's test.
For Hotelling-Lawley, Pillai, and Wilks, you can actually do a little better by multiplying by something a little different from the sample size (but asymptotically equivalent).
For Hotelling-Lawley

$$(f_e - p - 1)\sum_{i=1}^{p} \eta_i$$

For Wilks:

$$\left(f_e + \frac{f_h - p - 1}{2}\right)\sum_{i=1}^{p} \ln(1+\eta_i)$$

For Pillai:

$$(f_h + f_e)\sum_{i=1}^{p} \frac{\eta_i}{1+\eta_i}$$

```
Cmd> mvtest <- macro("
@h <- $1
@e <- $2
@fh <- $3
@fe <- $4
@p <- ncols(@h)
@vals <- releigen(@h,@e)$values
@lambda <- 1/prod(1+@vals)
@ht <- sum(@vals)
@pl <- sum(@vals/(1+@vals))
@r <- @vals[1]/(1+@vals[1])
@lambdap <- 1-cumchi( (@fe+(@fh-@p-1)/2) *\
 -log(@lambda),@fh*@p)
@htp <- 1-cumchi( (@fe-@p-1) * @ht, @fh*@p)
@plp <- 1-cumchi( (@fe+@fh)*@pl, @fh*@p)
@rlabs <- vector(\"Wilks lambda\",\
\"Hotelling-Lawley trace\",\"Pillai trace\",\
\"Roy’s largest root\")
@clabs <- vector(\"statistic\",\"p-value\")
@labs <- structure(@rlabs,@clabs)
@out <- matrix(vector(@lambda,@ht,@pl,@r,\
@lambdap,@htp,@plp,?),4,labels:@labs)
@out")

Cmd> mvtest(h,e,2,87)
                     statistic      p-value
Wilks lambda            0.8301     0.043531
Hotelling-Lawley       0.20188     0.035099
Pillai trace           0.17221     0.053093
Roy’s largest root     0.15752      MISSING
```

Now let’s try these on the extrusion data.

```
Cmd> manova("Y=a*b")
Model used is Y=a*b
...

Cmd> mvtest(SS[4,,],SS[5,,],1,16)
                     statistic      p-value
Wilks lambda           0.77711      0.30101
Hotelling-Lawley       0.28683      0.32837
Pillai trace           0.22289      0.28514
Roy’s largest root     0.22289      MISSING
```

Interaction not significant

```
Cmd> mvtest(SS[2,,],SS[5,,],1,16)
                     statistic      p-value
Wilks lambda           0.38186    0.0029612
Hotelling-Lawley        1.6188   0.00022327
Pillai trace           0.61814     0.014704
Roy's largest root     0.61814      MISSING

Cmd> mvtest(SS[3,,],SS[5,,],1,16)
                     statistic      p-value
Wilks lambda           0.52303     0.024447
Hotelling-Lawley       0.91192     0.012038
Pillai trace           0.47697     0.043824
Roy's largest root     0.47697      MISSING
```

Both main effects signifiant, more so for extrusion than additive.
Try testing all three degrees of freedom together; this is the total model fit.

```
Cmd> mvtest(SS[2,,]+SS[3,,]+SS[4,,],\
SS[5,,],3,16)
                     statistic      p-value
Wilks lambda           0.17802    0.0015378
Hotelling-Lawley        2.8175   9.6429e-05
Pillai trace            1.1456    0.0096497
Roy's largest root     0.65152      MISSING
```

Suppose that u_i is the ith eigenvector of $\mathbf{H}$ relative to $\mathbf{E}$.
u_1 is the combination of variables giving us the biggest "F" statistic. u_2 gives us the next biggests among vectors satisfying

$$u_1'\mathbf{H}u_2 = u_1'\mathbf{E}u_2 = 0$$

$\mathbf{Y}u_1$ and $\mathbf{Y}u_2$ reexpress the data along these canonical axes.
Plots along canonical axes can help show the multivariate differences.
Using the skull data

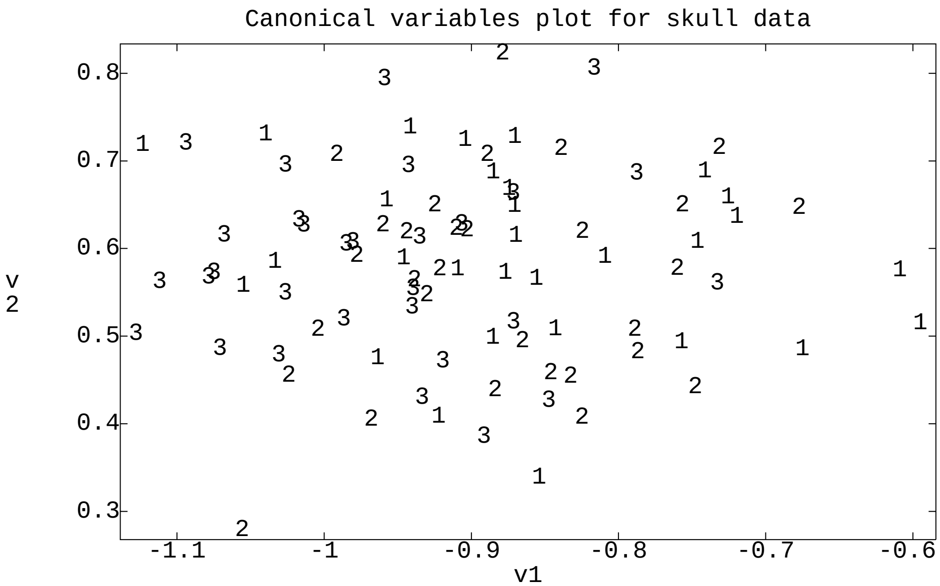
```
Cmd> vecs
            (1)          (2)          (3)          (4)
(1)    -0.01731     -0.01243 -0.0009532     0.011956
(2) -0.0031664     0.017222   -0.011639    0.0096574
(3)    0.015997  -0.0074584  -0.0013435     0.012517
(4)    0.004921     0.013185    0.032188    0.0067102

Cmd> vals
(1)     0.18697     0.014913           0            0
```

The eigenvalues suggest that we should see larger differences in the first rotated direction.

```
Cmd> v1 <- Y %*% vecs[,1]

Cmd> v2 <- Y %*% vecs[,2]

Cmd> tabs(v1,a,mean:T)
(1)        -0.862     -0.87954     -0.96626

Cmd> tabs(v2,a,mean:T)
(1)       0.59454       0.5634      0.58326
```

Yep, bigger differences in first direction.

```
Cmd> chplot(v1,v2,a,title:\
"Canonical variables plot for skull data")
```



Now extrusion data (combining all 3 treatment df).

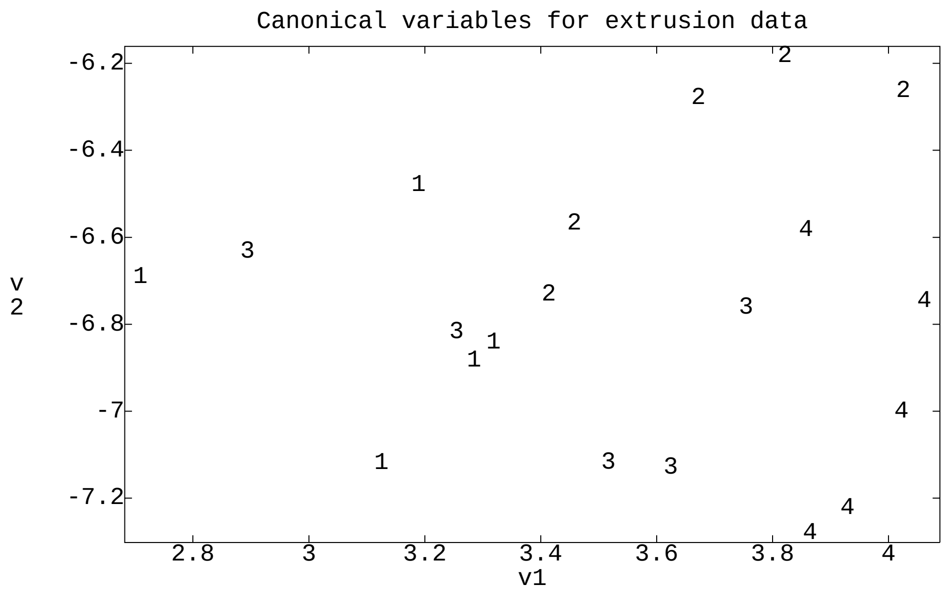
```
Cmd> manova("Y=a.b")
...

Cmd> h <- matrix(SS[2,,])

Cmd> e <- matrix(SS[3,,])

Cmd> vals <- releigen(h,e)$values;vals
(1)        1.8696      0.93765     0.010274

Cmd> vecs <- releigen(h,e)$vectors
```

Most of the differences are in the first two directions.

```
Cmd> v1 <- Y%*%vecs[,1]

Cmd> v2 <- Y%*%vecs[,2]

Cmd> ab <-a+(b-1)*2

Cmd> tabs(v1,ab,mean:T)
(1)    3.1255    3.6782     3.409    3.9473

Cmd> tabs(v2,ab,mean:T)
(1)    -6.803   -6.4052   -6.8916   -6.9666
```

Extrusion 2 is higher than ext 1 in the first direction. Additive 2 is lower than add 1 in the second direction.

```
Cmd> chplot(v1,v2,ab,title:\
"Canonical variables for extrusion data")
```



There are many more possible tests in MANOVA based on Bonferroni adjustments of univariate tests.
Univariate tests have the advantage that they only depend on univariate normality, and they may be more powerful when any differences are along the directions we choose to test.
The simplest thing to do is just test along original variables.
Let's use the extrusion data.

```
Cmd> manova("Y=a*b",byvar:T,pvals:T)
Model used is Y=a*b
WARNING: summaries are sequential
                         Variable 1
                 DF            SS            MS       P-value
CONSTANT          1        920.72        920.72             0
a                 1        1.7405        1.7405     0.0010917
b                 1        0.7605        0.7605       0.01833
```

```
a.b                1        0.0005       0.0005      0.94714
ERROR1            16         1.764      0.11025
                        Variable 2
                  DF            SS           MS      P-value
CONSTANT           1        1735.4       1735.4            0
a                  1        1.3005       1.3005     0.012479
b                  1        0.6125       0.6125     0.071387
a.b                1        0.5445       0.5445     0.087402
ERROR1            16         2.628      0.16425
                        Variable 3
                  DF            SS           MS      P-value
CONSTANT           1        309.68       309.68   1.7383e-07
a                  1        0.4205       0.4205      0.75169
b                  1        4.9005       4.9005      0.28805
a.b                1        3.9605       3.9605      0.33789
ERROR1            16        64.924       4.0578

Cmd> 3*min(.00109,.0125,.752)
(1)      0.00327

Cmd> 3*min(.01833,.0714,.288)
(1)      0.05499
```

Using Bonferroni adjusted univariate tests, extrusion is significant, but additive is only marginally significant.
We are not restricted to just using the original coordinates. We can use any set of *predetermined* linear combinations.
Note that we cannot use the relative eigen directions here, because those are determined by the data.
A good choice of directions should reflect knowledge of how we expect differences to occur.
Let's try the skull data. I'll use directions determined by

$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & -1 & 0 \\ 1 & -1 & 0 & 1 \\ 1 & -1 & 0 & -1 \end{bmatrix}$$

This considers the total, how the two large variables differ from the two small variables, the difference of the two large variables, and the difference of the two small variables.

```
Cmd> U <- matrix(vector(1,1,1,1,\
1,1,-1,-1,1,-1,0,0,0,0,1,-1),4);U
(1,1)       1       1       1       0
(2,1)       1       1      -1       0
(3,1)       1      -1       0       1
(4,1)       1      -1       0      -1

Cmd> Z <- Y %*% U

Cmd> manova("Z=a",byvar:T,pvals:T)
Model used is Z=a
```

```
WARNING: summaries are sequential
                        Variable 1
                DF            SS            MS       P-value
CONSTANT         1    1.5473e+07   1.5473e+07            0
a                2           3.8          1.9       0.98093
ERROR1          87        8583.1       98.656
                        Variable 2
                DF            SS            MS       P-value
CONSTANT         1     1.244e+06    1.244e+06            0
a                2         760.2        380.1     0.0036029
ERROR1          87        5505.9       63.286
                        Variable 3
                DF            SS            MS       P-value
CONSTANT         1          36.1         36.1       0.33665
a                2         130.2         65.1       0.19176
ERROR1          87        3364.7       38.675
                        Variable 4
                DF            SS            MS       P-value
CONSTANT         1     2.043e+05    2.043e+05            0
a                2        214.02       107.01      0.049742
ERROR1          87        2996.6       34.444
```

```
Cmd> 4*.0036
(1)       0.0144
```

A strongly significant difference between groups in how much larger the “big” variables are than the “small” variables.