THE UNIVERSITY OF MINNESOTA

Statistics 5401 October 10, 2005

Example of One-Way Multivariate Analysis of Variance

The data analyzed are the Fisher Iris data from Table 11.5 in Johnson and Wichern. Each observation consists of four quantities (sepal width and length, and petal width and length) measured on 50 blossoms from each of three varieties of Iris, *I. setosa*, *I. versicolor*, and *I. virginica* (Fisher, R. A. The use of multiple measurments in taxonomic problems *Ann. Eugen.* **7** (1936) 179-188).

Column 1 is variety number (1, 2, or 3), and columns 2 to 5 are X_1 = sepal length, X_2 = sepal width, X_3 = petal length, and X_4 = petal width.

One-way MANOVA using ANOVA on response variables

```
Cmd> y <- read("","t11_05",quiet:T) # read JWData5.txt
Read from file "TP1:Stat5401:Data:JWData5.txt"

Cmd> varieties <- y[,1] # get variety id

Cmd> # 1=setosa, 2=versicolor, 3=virginica

Cmd> y <- y[,-1] # remove varieties column

Cmd> N <- nrows(y)
```

Find residual SSCP matrix when H_0 is true

```
Cmd> ybar <- sum(y)/ N # ybar is grand mean row vector

Cmd> rcp0 <- (y - ybar)' %*% (y - ybar)
```

`ybar` is the mean of all the cases and is the best estimate of the common mean under the null hypotheses that all the means are zeros. Therefore `y - ybar` is the matrix of residuals and `rcp0` is the matrix of sums of squares and products of residuals, both computed assuming H_0 is true.

```
Cmd> print(rcp0) # Total matrix of SS and SP
rcp0:
             SepLen      SepWid      PetLen      PetWid
SepLen       102.17     -6.3227      189.87      76.924
SepWid      -6.3227      28.307     -49.119     -18.124
PetLen       189.87     -49.119      464.33      193.05
PetWid       76.924     -18.124      193.05       86.57
```

Find residual SSCP matrix when H_0 is false using univariate ANOVA

```
Cmd> # Do univariate ANOVA for each dependent variable

Cmd> varieties <- factor(varieties)

Cmd> anova("{y[,1]} = varieties",fstat:T) # ANOVA for sepal length
Model used is {y[,1]} = varieties
                 DF          SS          MS           F     P-value
CONSTANT          1      5121.7      5121.7 19326.50528     < 1e-08
varieties         2      63.212      31.606   119.26450     < 1e-08
ERROR1          147      38.956     0.26501

Cmd> residuals1 <- RESIDUALS # save sepal length residuals
```

```
Cmd> b1 <- coefs(); b1 # coefficients for sepal length
component: CONSTANT
(1)      5.8433
component: varieties
(1)    -0.83733    0.092667     0.74467 Variety  effects  alphahat
```

```
Cmd> anova("{y[,2]} = varieties",fstat:T) # ANOVA for sepal width
Model used is {y[,2]} = varieties
                DF          SS          MS           F     P-value
CONSTANT         1      1402.1      1402.1 12151.14260     < 1e-08
varieties        2      11.345      5.6725    49.16004     < 1e-08
ERROR1         147      16.962     0.11539
```

```
Cmd> residuals2 <- RESIDUALS # save sepal Width residuals
```

```
Cmd> b2 <- coefs(); b2 # coefficients for Sepal Width
component: CONSTANT
(1)      3.0573
component: varieties
(1)     0.37067    -0.28733   -0.083333 Variety effects
```

```
Cmd> anova("{y[,3]} = varieties",fstat:T) # ANOVA for petal length
Model used is {y[,3]} = varieties
                DF          SS          MS           F     P-value
CONSTANT         1      2118.4      2118.4 11439.11809     < 1e-08
varieties        2       437.1      218.55  1180.16118     < 1e-08
ERROR1         147      27.223     0.18519
```

```
Cmd> residuals3 <- RESIDUALS # save petal Length residuals
```

```
Cmd> b3 <- coefs(); b3 # coefficients for Petal Length
component: CONSTANT
(1)       3.758
component: varieties
(1)      -2.296       0.502       1.794 Variety  effects  alphahat
```

```
Cmd> anova("{y[,4]} = varieties",fstat:T) # ANOVA for petal width
Model used is {y[,4]} = varieties
                DF          SS          MS           F     P-value
CONSTANT         1      215.76      215.76  5151.66322     < 1e-08
varieties        2      80.413      40.207   960.00715     < 1e-08
ERROR1         147      6.1566    0.041882
```

```
Cmd> residuals4 <- RESIDUALS save petal width residuals
```

```
Cmd> b4 <- coefs(); b4 # coefficients for Petal Width
component: CONSTANT
(1)      1.1993
component: varieties
(1)    -0.95333     0.12667     0.82667 Variety  effects  alphahat
```

```
Cmd> residuals <- hconcat(residuals1,residuals2,residuals3,residuals4)
```

150 by 4 matrix `residuals` contains the ANOVA (least squares) residuals for all four variables. These are residuals computed with no restrictions on the parameters, that is assuming the alternative H_1 is true.

```
Cmd> setlabels(rcp1,structure(getlabels(y,2),getlabels(y,2)))

Cmd> rcp1 # residual SSCP matrix under H_1
            SepLen      SepWid      PetLen      PetWid
SepLen      38.956       13.63      24.625       5.645
SepWid       13.63      16.962      8.1208      4.8084
PetLen      24.625      8.1208      27.223      6.2718
PetWid       5.645      4.8084      6.2718      6.1566
```

This is the matrix of sums of squares and products of residuals computed without assuming H_0 is true.

Find the hypothesis matrix H and error matrix E

```
Cmd> h <- rcp0 - rcp1 # compute hypothesis matrix of SS & SP

Cmd> e <- rcp1 # compute error matrix of SS & SP

Cmd> print(h,e)
h:
            SepLen      SepWid      PetLen      PetWid
SepLen      63.212     -19.953      165.25      71.279
SepWid     -19.953      11.345      -57.24     -22.933
PetLen      165.25      -57.24       437.1      186.77
PetWid      71.279     -22.933      186.77      80.413
e:
            SepLen      SepWid      PetLen      PetWid
SepLen      38.956       13.63      24.625       5.645
SepWid       13.63      16.962      8.1208      4.8084
PetLen      24.625      8.1208      27.223      6.2718
PetWid       5.645      4.8084      6.2718      6.1566
```

One-way MANOVA by direct computation of SSCP matrices

```
Cmd> # Compute group mean vectors using macro groupcovar()

Cmd> stuff <- groupcovar(varieties,y)

Cmd> stuff # group sizes n, group means as rows of matrix
component: n
(1)          50          50          50
component: means
(1,1)       5.006       3.428       1.462       0.246
(2,1)       5.936        2.77        4.26       1.326
(3,1)       6.588       2.974       5.552       2.026
component: covariance
(1,1)     0.26501    0.092721     0.16751    0.038401
(2,1)    0.092721     0.11539    0.055244     0.03271
(3,1)     0.16751    0.055244     0.18519    0.042665
(4,1)    0.038401     0.03271    0.042665    0.041882
```

Compute H from vectors of means from formula for H in 1-way MANOVA

```
Cmd> h1 <- dmat(4,0) # alternative computation of h
Cmd> for(i,run(3)){ # accumulate h1 in loop over groups
         d <- vector(stuff$means[i,] - ybar)
         h1 <-+ stuff$n[i] * d %*% d';;
     }
Cmd> setlabels(h1,structure(getlabels(y,1),getlabels(y,1)))
```

Compute E from residuals from of group means

```
Cmd> res1 <- y[varieties==1,] - stuff$means[1,] # group 1 residuals
Cmd> res2 <- y[varieties==2,] - stuff$means[2,] # group 2 residuals
Cmd> res3 <- y[varieties==3,] - stuff$means[3,] # group 3 residuals
Cmd> list(res1,res2,res3)
res1            REAL   50    4
res2            REAL   50    4
res3            REAL   50    4
Cmd> residuals1 <- vconcat(res1, res2, res3) # all residuals
Cmd> list(residuals1)
residuals1      REAL   150   4
Cmd> e1 <- residuals1' %*% residuals1 #SS and SP matrix of residuals
Cmd> print(h1,e1) # same as before
h1:
            SepLen      SepWid      PetLen      PetWid
SepLen      63.212     -19.953      165.25      71.279
SepWid     -19.953      11.345      -57.24     -22.933
PetLen      165.25      -57.24       437.1      186.77
PetWid      71.279     -22.933      186.77      80.413
e1:
            SepLen      SepWid      PetLen      PetWid
SepLen      38.956       13.63      24.625       5.645
SepWid       13.63      16.962      8.1208      4.8084
PetLen      24.625      8.1208      27.223      6.2718
PetWid       5.645      4.8084      6.2718      6.1566
```

One-way MANOVA by regression on dummy variables

This time we do regression by a white box method, computing coefficients from $(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$ where column 1 of $\mathbf{X}$ is $\mathbf{1}_n$ and the remaining columns are contrast vectors comparing groups 1 and 2 with group 3.

```
Cmd> x1 <- (varieties == 1) - (varieties == 3)
Cmd> x2 <- (varieties == 2) - (varieties == 3)
```

`x1` has 1's for variety 1, 0 for variety 2 and –1 for variety 3. `x2` has 0's for variety 1, 1 for variety 2 and –1 for variety 3.

```
Cmd> hconcat(varieties, x1, x2)[vector(1,50,51,100,101,150),]
        Variety     Variety     Variety
(1)           1           1           0       Case   1 of group 1
(2)           1           1           0       Case  50 of group 1
(3)           2           0           1       Case   1 of group 2
(4)           2           0           1       Case  50 of group 2
(5)           3          -1          -1       Case   1 of group 3
(6)           3          -1          -1       Case  50 of group 3

Cmd> X <- hconcat(rep(1,N), x1, x2)# constant vector & dummies matrix

Cmd> bhat <- solve(X' %*% X, X' %*% y); bhat #regression coefficients
(1,1)      5.8433      3.0573       3.758      1.1993  mu
(2,1)    -0.83733     0.37067      -2.296    -0.95333  alphahat_1
(3,1)    0.092667    -0.28733       0.502     0.12667  alphahat_2
```

Compare these with the values computed earlier using `anova()`.

```
Cmd> vector(b1$CONSTANT, b2$CONSTANT, b3$CONSTANT, b4$CONSTANT)
(1)      5.8433      3.0573       3.758      1.1993    μ

Cmd> hconcat(b1$varieties, b2$varieties, b3$varieties, b4$varieties)
(1,1)    -0.83733     0.37067      -2.296    -0.95333  alphahat_1
(2,1)    0.092667    -0.28733       0.502     0.12667  alphahat_2
(3,1)     0.74467   -0.083333       1.794     0.82667  alphahat_3
```

This has a row for group 3 consisting of the estimates of $\alpha_3 = -\alpha_1 - \alpha_2$. These effects sum to zero accross groups.

```
Cmd> -sum(bhat[-1,]) # same as last line of preceding
(1,1)     0.74467   -0.083333       1.794     0.82667

Cmd> residuals2 <- y - X %*% bhat # regression residuals

Cmd> e2 <- residuals2' %*% residuals2; e2 # same as before
            SepLen      SepWid      PetLen      PetWid
SepLen      38.956       13.63      24.625       5.645
SepWid       13.63      16.962      8.1208      4.8084
PetLen      24.625      8.1208      27.223      6.2718
PetWid       5.645      4.8084      6.2718      6.1566
```

Computation of Test Statistics Based on Relative Eigenvalues

```
Cmd> eigs <- releigen(h,e); eigs # compute relative eigen things
component: values             Relative eigenvalues, last 2 are 0
(1)      32.192     0.28539  7.8772e-15 -4.7485e-16
component: vectors            Relative eigenvectors
(               (1)         (2)         (3)         (4)
SepLen   -0.068406   0.0019879     0.14375     0.22004
SepWid    -0.12656     0.17853    -0.20981   -0.070133
PetLen     0.18155   -0.076864    -0.24008   -0.064018
PetWid      0.2318     0.23417     0.37038   -0.066357

Cmd> # Note that only 2 eigenvalues are essentially non-zero,

Cmd> # reflecting the fact that fh (hypothesis d.f) is 2.
```

```
Cmd> round(eigs$vectors' %*% h %*% eigs$vectors,10)#round to 10 dec
             (1)         (2)         (3)         (4)
(1)       32.192           0           0           0
(2)            0     0.28539           0           0
(3)            0           0           0           0
(4)            0           0           0           0

Cmd> # Confirms that U'HU is dmat(eigs$values), U eig vector matrix

Cmd> # Note round(x,n) rounds x to n decimal places

Cmd> round(eigs$vectors' %*% e %*% eigs$vectors, 10)
             (1)         (2)         (3)         (4)
(1)            1           0           0           0
(2)            0           1           0           0
(3)            0           0           1           0
(4)            0           0           0           1

Cmd> # and U'EU is I4

Cmd> maxroot <- eigs$values[1]; maxtheta <- maxroot/(1 + maxroot)

Cmd> vector(maxroot,maxtheta)
(1)       32.192     0.96987
```

$f_e\lambda_{max}/f_h$ is the largest among group F-statistic for test equality of means for any linear combination of response variables. The coefficients of the maximizing linear combination are the elements of the corresponding eigenvector. The following demonstrates this numerically.

```
Cmd> maxf <- fe*maxroot/fh ; maxf
(1,1)       2366.1

Cmd> # This is the largest possible univariate F for any linear combo

Cmd> anova("{y %*% u1} = varieties",fstat:T,pvals:F)
Model used is {y %*% u1} = varieties
                 DF          SS          MS           F
CONSTANT          1      4.5219      4.5219   664.72097
varieties         2      32.192      16.096  2366.10680
ERROR1          147           1   0.0068027
```

An approximate critical value for $\theta_{max} = \lambda_{max}/(1 + \lambda_{max})$ can be read off from charts included in another handout. They are entered with $s = \min(p, f_h)$, $m = (|f_h - p| - 1)/2$ and $n = (f_e - p - 1)/2$.

```
Cmd> g <- max(varieties); g # number of groups
(1,1)            3

Cmd> fh <- g - 1; fe <- (N - 1) - fh # 1 for constant term

Cmd> vector(fh, fe)
(1)            2         147

Cmd> p <- ncols(y) # number of variables

Cmd> s <- min(vector(p,fh)); m <- (abs(fh - p) - 1)/2

Cmd> n <- (fe-p-1)/2 # this is not sample size

Cmd> vector(s,m,n) # the 3 quantities needed for using charts
(1)            2         0.5          71
```

The approximate 1% critical value for θ_{max} from the chart for s = 2, m = 1/2 and n = 71 is 0.113, *much* less than θ_{max} = 0.970. H_0 is strongly rejected.

Test of H_0 by Bonferronized univariate F-tests

```
Cmd> # diagonals of h and e are univariate hypothesis and error SS
Cmd> f <- vector((diag(h)/fh)/(diag(e)/fe)); f # univariate f's
(1)      119.26       49.16      1180.2      960.01
Cmd> p*(1 - cumF(f,fh,fe)) # Bonferronized P-values
(1)            0            0            0            0
Cmd> # All highly significant
```

Test of H_0 by Hotelling's generalized T_0^2

```
Cmd> # Now compute Hotelling's generalized T0^2 (trace statistic)
Cmd> t0sq <- fe*trace(solve(e) %*% h); t0sq # as fe*trace e^(-1) h
(1,1)      4774.2           From H and E matrices
Cmd> fe*sum(eigs$values) # or as sum of relative eigenvalues
(1,1)      4774.2           From relative eigenvalues
```

In large samples, $(f_e - p - 1)T_0^2/f_e$ is approximately $\chi_{pf_h}^2$.

```
Cmd> m2 <- fe - p - 1
Cmd> cumchi(m2*t0sq/fe, fh*p, upper:T)
(1)           0        Rediculously small P-value
```

Although it doesn't matter here because **H** is so large relative to **E**, you can find a more exact P-value using macro `cumtrace()` in the new version of macro file `Mulvar.mac`.

```
Cmd> addmacrofile("") # find new version of mulvar.mac
Cmd> cumtrace(trace(solve(e,h)),fh,fe,p,upper:T)
(1)           0
```

Test of H_0 by likelihood ratio statistic Λ^*

```
Cmd> lambdastar <- det(solve(h+e,e)); lambdastar = Wilk's statistic
(1)    0.023439         From H and E matrices
Cmd> det(e)/det(h+e)
(1)    0.023439         Another way from H and E
Cmd> 1/prod(1 + eigs$values)
(1)    0.023439         From relative eigenvalues
Cmd> m1 <- fe - (p - fh + 1)/2; m1 # adjusting constant
(1)       145.5
Cmd> # Compute corrected log likelihood ratio test, with null
Cmd> # distribution approximately chisq(fh*p) = chisq(8)
Cmd> lrtest <- -m1*log(lambdastar); lrtest
(1,1)      546.12
Cmd> # A somewhat more accurate approximation due to Rao uses F
Cmd> approxf <- ((fe-p+1)/p)*(1 - sqrt(lambdastar))/sqrt(lambdastar)
```

```
Cmd> approxf # Distributed when min(p,fh)=2 as F(2*p,2*(fe-p+1))
(1,1)      199.15
Cmd> vector(2*p,2*(fe-p+1)) # degrees of freedom for approximate F
(1)           8         288
Cmd> cumF(approxf,2*p,2*(fe-p+1),upper:T) # P-value, highly significant
(1)  1.365e-112
```

Actually, since min(p, f_h) = 2, Rao's F is exact.

There is a black box way to compute this using `cumwilks()` from new version of `Mulvar.mac`.

```
Cmd> cumwilks(lambdastar,fh,fe,p,all:T)
component: P
(1)  1.365e-112
component: f
(1)      199.15
component: df1
(1)           8
component: df2
(1)         288
```

Without `all:T` you get just the P-value:

```
Cmd> cumwilks(lambdastar,fh,fe,p)
(1)  1.365e-112
```

Test of H_0 by Pillai's trace statistic

```
Cmd> pillai <- trace(solve(h+e,h)); pillai
(1)      1.1919
Cmd> sum(vals/(1+vals)) # same thing from relative eignevalues
(1)      1.1919
Cmd> m3 <- fe + fh # multiplier for Pillai's trace statistic V
Cmd> v <- m3*pillai;v
(1)      177.59
```

In large samples, when H_0 is true, V is approximately $\chi_{pf_h}^2$

```
Cmd> cumchi(v,fh*p,upper:T)
(1)  3.2959e-34
```

Although it's not really needed here, you can get a more accurate P-value using macro `cumpillai()` in the new version of file `Mulvar.mac`.

```
Cmd> cumpillai(pillai, fh,fe,p,upper:T)
(1)  1.7638e-31
```

One-Way MANOVA using the `manova()` command

```
Cmd> varieties <- factor(varieties) # already done above
Cmd> # preceding ensures that varieties interpreted as factor levels
```

```
Cmd> manova("y=varieties") # simplest 1-way MANOVA model
WARNING: summaries are sequential
                    SS and SP Matrices
                 DF
CONSTANT          1
            SepLen      SepWid      PetLen      PetWid
SepLen      5121.7      2679.8      3293.9      1051.2
SepWid      2679.8      1402.1      1723.4      550.01
PetLen      3293.9      1723.4      2118.4      676.06
PetWid      1051.2      550.01      676.06      215.76
varieties        2          Hypothesis matrix H
            SepLen      SepWid      PetLen      PetWid
SepLen      63.212     -19.953      165.25      71.279
SepWid     -19.953      11.345      -57.24     -22.933
PetLen      165.25      -57.24       437.1      186.77
PetWid      71.279     -22.933      186.77      80.413
ERROR1         147          Error matrix E
            SepLen      SepWid      PetLen      PetWid
SepLen      38.956       13.63      24.625       5.645
SepWid       13.63      16.962      8.1208      4.8084
PetLen      24.625      8.1208      27.223      6.2718
PetWid       5.645      4.8084      6.2718      6.1566

Cmd> list(SS,DF,TERMNAMES,RESIDUALS,HII)
DF              REAL   3     (labels)
HII             REAL   150   (labels)
RESIDUALS       REAL   150   4     (labels)
SS              REAL   3     4     4     (labels)
TERMNAMES       CHAR   3

Cmd> # These are "side effect" arrays and vectors created by manova()

Cmd> SS # notice it is automatically labelled with term names
                      SepLen      SepWid      PetLen      PetWid
CONSTANT  SepLen      5121.7      2679.8      3293.9      1051.2
          SepWid      2679.8      1402.1      1723.4      550.01
          PetLen      3293.9      1723.4      2118.4      676.06
          PetWid      1051.2      550.01      676.06      215.76
varieties SepLen      63.212     -19.953      165.25      71.279
          SepWid     -19.953      11.345      -57.24     -22.933
          PetLen      165.25      -57.24       437.1      186.77
          PetWid      71.279     -22.933      186.77      80.413
ERROR1    SepLen      38.956       13.63      24.625       5.645
          SepWid       13.63      16.962      8.1208      4.8084
          PetLen      24.625      8.1208      27.223      6.2718
          PetWid       5.645      4.8084      6.2718      6.1566

Cmd> print(DF, TERMNAMES)
DF:         Degrees of freedom for each term
    CONSTANT   varieties      ERROR1
           1           2         147
TERMNAMES: Names for each term
(1) "CONSTANT"
(2) "varieties"
(3) "ERROR1"
```

```
Cmd> fh <- DF[2]; fe <- DF[3]; vector(fh, fe)
   varieties      ERROR1
           2         147

Cmd> releigen(SS[2,,],SS[3,,]) # again same results
component: values
(1)      32.192     0.28539  7.5171e-15  1.7978e-16
component: vectors
               (1)         (2)         (3)         (4)
SepLen   -0.068406   0.0019879     0.13261     0.22693
SepWid    -0.12656     0.17853    -0.20606   -0.080491
PetLen     0.18155   -0.076864     -0.2366    -0.07589
PetWid      0.2318     0.23417     0.37322   -0.047837

Cmd> h <- matrix(SS[2,,]); e <- matrix(SS[3,,])

Cmd> fh <- DF[2]; fe <- DF[3], p <- nrows(h)

Cmd> cumwilks(det(e)/det(h+e),fh,fe,p)
(1)  1.365e-112
```

MANOVA Canonical Variables

An important use of the eigenvectors $\hat{\mathbf{u}}_j$ of **H** relative to **E** is in computing the estimated MANOVA canonical variables $\hat{z}_j \equiv \hat{\mathbf{u}}_j'\mathbf{y}$, associated with the null hypothesis H_0 being tested, where **y** is the p by 1 vector of response variables. If $\hat{\mathbf{U}} = \left[\hat{\mathbf{u}}_1\, \hat{\mathbf{u}}_2\, \hat{\mathbf{u}}_3 \ldots \hat{\mathbf{u}}_p\right]$ is the matrix of eigenvectors (`releigen(h,e)$vectors`), the estimated canonical variables are the columns of the n by p matrix $\mathbf{Z} = \mathbf{Y}\hat{\mathbf{U}}$. Thus the ith canonical variable is $\hat{z}_i = \hat{\mathbf{u}}_i'\mathbf{y} = \mathbf{y}'\hat{\mathbf{u}}_i$, a linear combination of the response variables.

If you compute a p by p hypothesis matrix $\mathbf{H}_{zz}$ and an error matrix $\mathbf{E}_{zz}$ treating the canonical variables as data, then $\mathbf{H}_{zz} = \hat{\mathbf{U}}'\mathbf{H}\hat{\mathbf{U}} = \text{diag}[\hat{\lambda}_1, \hat{\lambda}_2, \ldots, \hat{\lambda}_p]$ and $\mathbf{E}_{zz} = \mathbf{I}_p$. Thus the canonical variables are uncorrelated and the univariate F-statistic for testing H_0, using only $\hat{z}_j$ is $(f_e / f_h)\hat{\lambda}_j$.

This can be interpreted as saying that $\hat{z}_1$ is the linear combination of the original variables for which H_0 is "most violated," that is, has the largest F-statistic. Similarly, $\hat{z}_2$ is the linear combination for which H_0 is "most violated" among all linear combinations *that are uncorrelated with* $\hat{z}_1$. etc.

For j > s = min(f_h,p), $\hat{z}_j = 0$ and the F-statistic based on $\hat{z}_j$ is zero, that is, there is *no* deviation from H_0 – all the sample group means of $\hat{z}_j$ are *identical*. In a sense, the first s canonical variables have "extracted" all the information concerning possible violations of H_0

It is worth repeating that *a particular set of canonical variables is associated with a particular null hypothesis*. In an anlysis with several hypothesis tests based on hypothesis matrices $\mathbf{H}_{01}$, $\mathbf{H}_{02}$, ..., there will be *different* sets of canonical variables associated with each $\mathbf{H}_{0i}$.

It can be informative to plot the first few canonical variables associated with a null hypothesis against each other. You often see much greater delineation of group structure in canonical variable plots than in plots of the original data. Furthermore, the

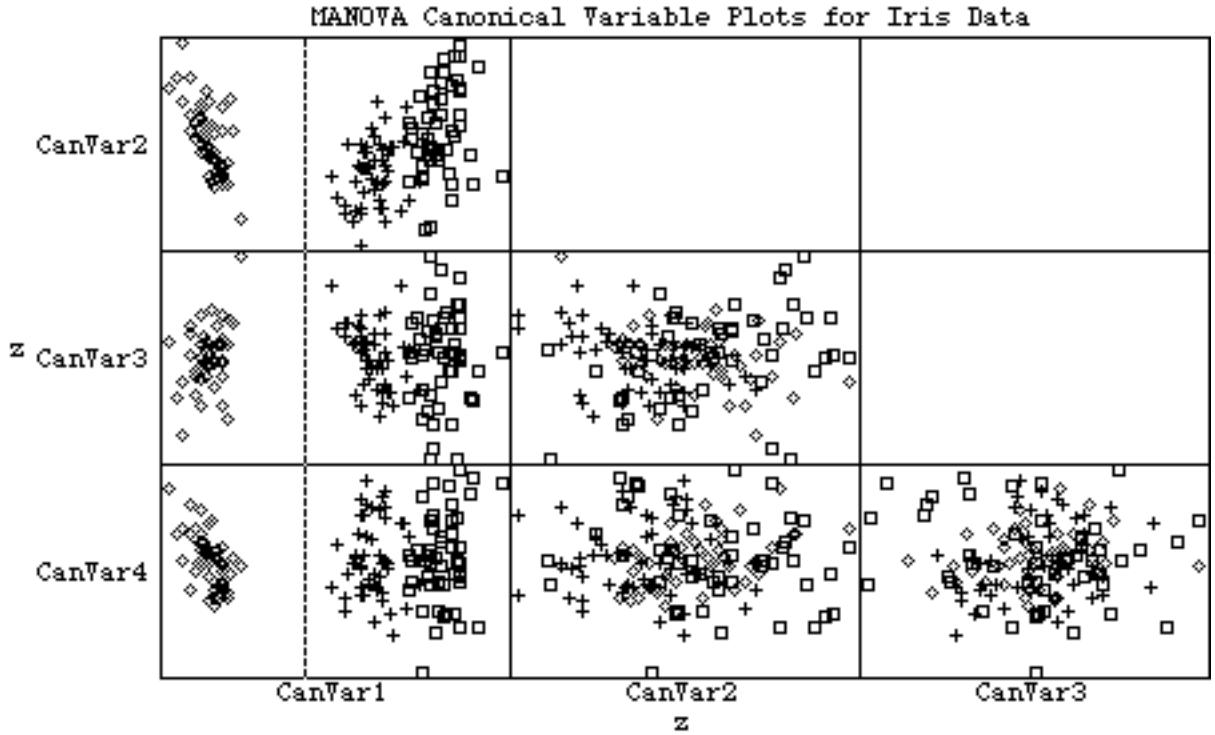
coefficients in these linear combinations, that is, the elements of the eigenvectors, sometimes provide information as to how the overall H_0 is violated.

```
Cmd> # Make "canonical variable" plots
Cmd> manova("y=varieties",silent:T) # start from the beginning
Cmd> h <- matrix(SS[2,,]); e <- matrix(SS[3,,])
Cmd> eigs <- releigen(h,e)
Cmd> u <- eigs$vectors ; print(u)
u: Coefficients of canonical variables
               (1)         (2)         (3)         (4)
SepLen   -0.068406   0.0019879     0.13261     0.22693
SepWid    -0.12656     0.17853    -0.20606   -0.080491
PetLen     0.18155   -0.076864     -0.2366    -0.07589
PetWid      0.2318     0.23417     0.37322   -0.047837
```

The columns of u are the eigenvectors of **H** relative to **E** and the elements in row i are the coefficients of variable i in computing the canonical variables.

```
Cmd> # compute values of all 4 estimated canonical vars
Cmd> z <- y %*% u # z is 150 by 4
Cmd> # Each column of z is linear combination of columns of y
Cmd> manova("z=varieties",silent:T) # do MANOVA on Z's
Cmd> round(SS[2,,],12) # Hzz = diag(eigenvalues); Ezz = I4
                      (1)         (2)         (3)         (4)
varieties (1)      32.192           0           0           0
          (2)          0     0.28539           0           0
          (3)          0           0           0           0
          (4)          0           0           0           0
```

```
Cmd> symbols <- vector("\11","\12","\13")[varieties] # medium sized
Cmd> plotmatrix(z,symbols:symbols,lower:T,\
title:"MANOVA Canonical Variable Plots for Iris Data",labels:"CanVar")
```




```
Cmd> # Compare these plots with some plots of the original data
Cmd> plotmatrix(y,lower:T,\
     title:"Fisher Iris Data",symbols:symbols,xlab:" ",ylab:" ")
```

Testing homogeneity of variance matrices

An important assumption underlying MANOVA is that the within group variance matrices are equal, that is, $\mathbf{\Sigma}_1 = \mathbf{\Sigma}_2 = \ldots = \mathbf{\Sigma}_g$. Assuming the multivariate normality of each population, an adjusted form of the log likelihood ratio test statistic to test H_0: $\mathbf{\Sigma}_1 = \mathbf{\Sigma}_2 = \ldots = \mathbf{\Sigma}_g$ against H_1: $\mathbf{\Sigma}_i \neq \mathbf{\Sigma}_j$ for some $i \neq j$, is

$$M = -C_d \times f_e \times \ln \tilde{\Lambda}_d = C_d \times \left\{ \ln \det(S) - \frac{1}{f_e} \sum_i f_{e_i} \ln \det(S_i) \right\},$$

where

$$C_d \equiv 1 - \frac{2p^2 + 3p - 1}{6(p+1)(g-1)} \left(\sum_i \frac{1}{f_{e_i}} - \frac{1}{f_e} \right)$$

Under the null hypothesis of equal $\mathbf{\Sigma}$'s, M is approximately distributed as χ_f^2, where $f = p(p+1)(g-1)/2$. This is a generalization of Bartlett's test for univariate homogeneity of variance. Like Bartlett's test, it is not at all robust to departures from normality. That is, when the populations are *not* normal the distribution of M under H_0 may very poorly approximated by χ_f^2. Here is an example of the computation of M to check variance matrix equality.

```
Cmd> S <- tabs(y,varieties,covar:T) # compute all within group covs
Cmd> S # 3 dimension array
(1,1,1)     0.12425    0.099216    0.016355    0.010331        s1
(1,2,1)    0.099216     0.14369    0.011698    0.009298
(1,3,1)    0.016355    0.011698    0.030159   0.0060694
(1,4,1)    0.010331    0.009298   0.0060694    0.011106
(2,1,1)     0.26643    0.085184      0.1829     0.05578        s2
(2,2,1)    0.085184    0.098469    0.082653    0.041204
(2,3,1)      0.1829    0.082653     0.22082    0.073102
(2,4,1)     0.05578    0.041204    0.073102    0.039106
(3,1,1)     0.40434    0.093763     0.30329    0.049094        s3
(3,2,1)    0.093763       0.104     0.07138    0.047629
(3,3,1)     0.30329     0.07138     0.30459    0.048824
(3,4,1)    0.049094    0.047629    0.048824    0.075433
Cmd> s1 <- matrix(S[1,,]); s2 <- matrix(S[2,,]); s3 <- matrix(S[3,,])
Cmd> print(s1,s2,s3)
s1:          Variance matrix for variety 1
(1,1)     0.12425    0.099216    0.016355    0.010331
(2,1)    0.099216     0.14369    0.011698    0.009298
(3,1)    0.016355    0.011698    0.030159   0.0060694
(4,1)    0.010331    0.009298   0.0060694    0.011106
s2:          Variance matrix for variety 2
(1,1)     0.26643    0.085184      0.1829     0.05578
(2,1)    0.085184    0.098469    0.082653    0.041204
(3,1)      0.1829    0.082653     0.22082    0.073102
(4,1)     0.05578    0.041204    0.073102    0.039106
```

```
s3:          Variance matrix for variety 3
(1,1)     0.40434    0.093763     0.30329    0.049094
(2,1)    0.093763       0.104     0.07138    0.047629
(3,1)     0.30329     0.07138     0.30459    0.048824
(4,1)    0.049094    0.047629    0.048824    0.075433

Cmd> p <- ncols(y); g <- 3;fe1 <- 49;fe2 <- 49;fe3 <- 49

Cmd> fe <- nrows(y) - g # or fe1 + fe2 + fe3

Cmd> spooled <- (fe1*s1 + fe2*s2 + fe3*s3)/fe # or e/df

Cmd> print(spooled,fe*spooled) # note fe*spooled is same as e
spooled:
(1,1)     0.26501    0.092721     0.16751    0.038401
(2,1)    0.092721     0.11539    0.055244     0.03271
(3,1)     0.16751    0.055244     0.18519    0.042665
(4,1)    0.038401     0.03271    0.042665    0.041882
MATRIX:   fe*spooled
(1,1)      38.956       13.63      24.625       5.645
(2,1)       13.63      16.962      8.1208      4.8084
(3,1)      24.625      8.1208      27.223      6.2718
(4,1)       5.645      4.8084      6.2718      6.1566

Cmd> bartlett <- fe*(log(det(spooled)) - \
     (fe1*log(det(s1)) + fe2*log(det(s2)) + fe3*log(det(s3)))/fe)

Cmd> f <- p*(p+1)*(g-1)/2; vector(bartlett,f)# Raw test statistic, df
(1)      146.66          20

Cmd> cd <- 1-((2*p*p+3*p-1)/(6*(p+1)*(g-1)))*(1/fe1+1/fe2+1/fe3-1/fe)

Cmd> cd # correction factor
(1)        0.961

Cmd> vector(cd*bartlett, 1-cumchi(cd*bartlett,f))# highly significant
(1)       140.94             0 Test statistic and P-value
```