

## Multi-group Profile Analysis Example

This handout provides an analysis of some artificial data from Example 5.9 on p. 240 of *Multivariate Statistical Methods*, 3<sup>rd</sup> Edition by Donald F. Morrison, McGraw Hill 1990.

Each observation represents one subject's scores on three scales A, B and C computed from a test instrument. Each subject was classified in one of four socioeconomic classes, 1, 2, 3 and 4 with sample sizes  $n_1 = 8$ ,  $n_2 = 5$ ,  $n_3 = 4$  and  $n_4 = 4$ , respectively. The data from Morrison's Table 5.7 are in data set TAB5.8 in file cbmorex.txt (5.8 was the table number in the second edition).

There are three columns of contrast dummy vectors but no table of groups numbers. Thus the first thing I did was to build factor groups from these dummy vectors.

```

Cmd> y <- read("", "TAB5.8") # read from cbmorex.txt
TAB5.8          21          6 FORMAT
) Data from Table 5.8, p. 210 of Morrison
) Col. 1: c1 = dummy variable (1,0,0,-1) for class 1
) Col. 2: c2 = dummy variable (0,1,0,-1) for class 2
) Col. 3: c3 = dummy variable (0,0,1,-1) for class 3
) Col. 4-6: a,b,c = scores on scales a,b,c
) n1=8,n2=5,n3=4,n4=4
Read from file "TP1:Stat5401:Data:cbmorex.txt"

Cmd> groups <-\    Construct factor from dummy variables
      factor(1*(y[,1]==1)+2*(y[,2]==1)+3*(y[,3]==1)+4*(y[,1]==-1))

Cmd> print(format:"1.0f",vector(groups)) # make sure we have it right
VECTOR:
(1) 1 1 1 1 1 1 1 1 2 2 2 2 2 3 3 3 3 4 4 4 4

Cmd> y <- y[, -run(3)] # trim off dummy variables
Cmd> setlabels(y, structure("@", vector("A", "B", "C")))
Cmd> list(y)
y              REAL    21    3    (labels)

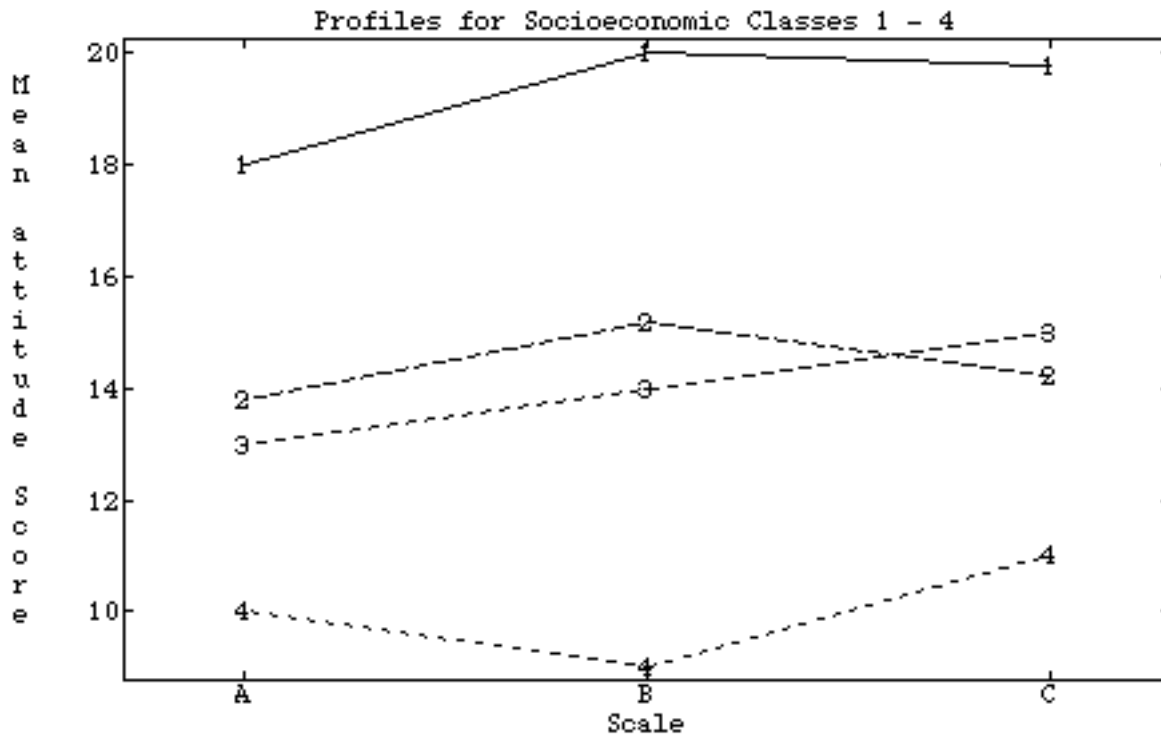
Cmd> stats <- tabs(y, groups, covar:T, mean:T, n:T)
Cmd> compnames(stats) # names of components of structure stats
(1) "mean"
(2) "covar"
(3) "count"

Cmd> stats$mean # each row is a group mean
(1,1)          18          20          19.75 Group 1
(2,1)          13.8          15.2          14.2 Group 2
(3,1)          13           14           15 Group 3
(4,1)          10           9            11 Group 4

```

## Profile Analysis Example

```
Cmd> # Make a plot of the group profiles rowplot()
Cmd> # If the means were in columns you would use colplot()
Cmd> rowplot(stats$mean,xticks:run(3),xmin:.75,xmax:3.25,\
  title:"Profiles for Socioeconomic Classes 1 - 4",\
  xticklabs:getlabels(y,2).\
  xlab:"Scale",ylab:"Mean attitude Score")
```



It appears there is a very substantial difference between groups and less difference between scales but it is not clear whether the lack of parallelism of the profiles is significant.

First load the new macros for computing P-values of multivariate tests.

```
Cmd> getmacros(cumwilks,cumtrace,cumpillai,quiet:T)
cumwilks read from file "TP1:Stat5401:Stat5401F04:Macros:Mulvar.mac"
cumtrace read from file "TP1:Stat5401:Stat5401F04:Macros:Mulvar.mac"
cumpillai read from file "TP1:Stat5401:Stat5401F04:Macros:Mulvar.mac"
```

The do MANOVA ignoring the repeated measures aspect of the data.

```
Cmd> manova("y=groups") # compute a MANOVA of the data
Model used is y=groups This ignores profile considerations
WARNING: summaries are sequential
      SS and SP Matrices
```

		DF		
		1		
CONSTANT		A	B	C
A	4429.8	4763.8	4836.4	
B	4763.8	5123	5201.1	
C	4836.4	5201.1	5280.4	

## Profile Analysis Example

groups	3		
	A	B	C
A	190.44	252.99	207.37
B	252.99	340.15	274.06
C	207.37	274.06	232.27

	17		
	A	B	C
A	42.8	11.2	18.2
B	11.2	30.8	3.8
C	18.2	3.8	38.3

```

Cmd> h <- matrix(SS[2,,]); e <- matrix(SS[3,,])
Cmd> fh <- DF[2]; fe <- DF[3]; p <- ncols(y)
Cmd> vector(fh, fe, p) # degrees of freedom and dimension
(1)          3          17          3

```

First look at difference between groups by regular MANOVA without regard to the repeated measurement aspect of the data.

```

Cmd> vals <- releigenvals(h,e);vals # relative eigenvalues
(1)      15.375      0.23073      0.035694

Cmd> theta <- vals/(1 + vals); theta
(1)      0.93893      0.18747      0.034464

Cmd> s <- min(fh,p); m <- (abs(fh-p)-1)/2; n <- (fe-p-1)/2
Cmd> vector(s,m,n)
(1)          3          -0.5          6.5

```

From the  $\alpha = .05$ ,  $s = 3$  chart, the critical value for  $\theta_{\max}$  from is about  $.51 << 0.93893$  so the overall between group differences in the mean vectors are very significant.

Let's test the same hypothesis using the other three tests, using macros `cumwilks()`, `cumtrace()` and `cumpillai()` to get P-values.

```

Cmd> cumwilks(det(e)/det(h+e),fh,fe,p) # P-value for LR test
(1)  1.2879e-07      Very highly significant

Cmd> cumtrace(trace(solve(e,h)),fh,fe,p,upper:T) # Hottelling trace
(1)  7.1343e-34      Ditto

Cmd> cumpillai(trace(solve(e+h,h)),fh,fe,p,upper:T) # Pillai's trace
(1)  -0.000687      Ooops! Bug in cumpillai()?

```

Now let's explore how the differences we have found can be described. You can compute profile analysis quantities from what we already have computed.

```

Cmd> c <- matrix(vector(1,-1,0, 0,1,-1),3) '#contrast matrix
Cmd> setlabels(c,structure(vector("AvsB","CvsB"),getlabels(y,2))); c
      A      B      C
AvsB  1     -1      0
CvsB  0      1     -1

Cmd> chc <- c %*% h %*% c' # hypothesis matrix for contrasts
Cmd> cec <- c %*% e %*% c' # error matrix for contrasts

```

## Profile Analysis Example

```

Cmd> print(chc,cec)
chc: Hypothesis matrix for parallelism (no-interaction)
      AvsB      CvsB
AvsB      24.61     -20.476
CvsB     -20.476      24.31
cec: Error matrix for parallelism (no-interaction)
      AvsB      CvsB
AvsB      51.2      -34
CvsB     -34       61.5

Cmd> val <- releigenvals(chc, cec);val # new relative eigenvalues
(1)      0.50885      0.17649

Cmd> theta <- val/(1+val); theta
(1)      0.33724      0.15002

Cmd> q <- nrow(chc) # new dimension
Cmd> vector(fh, fe, q) # fe and fh are the same, dimension is reduced
(1)           3           17           2

Cmd> s <- min(fh,q);m <- (abs(fh-q)-1)/2; n <- (fe-q-1)/2
Cmd> vector(s,m,n)
(1)           2           0           7

Cmd> # From the chart for alpha = .05, the critical value for
Cmd> # thetamax is about 0.47 > 0.337. We cannot reject parallelism
Cmd> # Just for illustration, we do other tests of H0 based on
Cmd> # relative eigenvalues based on handout on MANOVA tests
Cmd> m1 <- fe-(q-fh+1)/2; vector(m1,2*n+m+s+1) # adjstment to LR test
(1)           17           17 Both formulas give same m1

Cmd> wilks <- m1*sum(log(1 + val))
Cmd> vector(q*fh,wilks,cumchi(wilks,q*fh,upper:T))
(1)           6           9.7561      0.13531 DF, test stat, P-value

Cmd> # P-value is .13531; same conclusion
Cmd> cumwilks(1/prod(1 + val),fh,fe,q)
(1)      0.13644 Exact P-value, same conclusion

Cmd> m2 <- fe - (q+1); vector(m2, 2*n) #adjustment for T0sq
(1)           14           14 Both formulas give same m2

Cmd> hotelling <- m2*sum(val)
Cmd> vector(q*fh,hotelling,cumchi(hotelling,q*fh,upper:T))
(1)           6           9.5948      0.14279 DF, test stat, P-value

Cmd> # large sample P-value is .14279; same conclusion
Cmd> cumtrace(sum(val),fh,fe,q,upper:T) # Hottelling
(1)      0.15187 Close to exact; same conclusion

Cmd> m3 <- fe + fh;vector(m3, 2*(m+n+s+1)) # Constant for Pillai's V
(1)           20           20 Both formulas give same m3

Cmd> pillaiV <- m3*sum(theta)

```

## Profile Analysis Example

```

Cmd> vector(q*fh,pillaiV,cumchi(pillaiV,q*fh,upper:T)) #large sample
(1)          6          9.7452          0.1358 DF, test stat, P-value
Cmd> # large sample P-value is 0.1358; same conclusion
Cmd> cumpillai(sum(val/(1+val)),fh,fe,q,upper:T)
(1)          0.11612          Close to exact; same conclusion

```

The conclusion means there is no substantial statistical evidence that the profiles are not parallel, that is there does not appear to be interaction between the among subject factor, socio-economic class, and the within subject factor, test instrument scale.

Now look at among-scale main effects, *assuming parallelism*. You test whether contrasts in the overall means ignoring groups are 0 using Hotelling's  $T^2$ .

```

Cmd> grandmean <- describe(y,mean:T); grandmean
(1)          14.524          15.619          15.857
Cmd> N <- nrows(y) # Total Sample size
Cmd> vhat <- (e/fe)*(1/N) #estimated variance matrix of grandmean
Cmd> cybar <- c %%% grandmean; cvhatc <- c %%% vhat %%% c'
Cmd> tsq <- cybar' %%% solve(cvhatc) %%% cybar
Cmd> tsq #Hotelling's T^2
(1)          16.913
Cmd> f <- (fe - q + 1)*tsq/(fe*q);f # corresponding F-statistic
(1,1)          7.9588
Cmd> cumF(f,q,fe - q + 1,upper:T) # P- value using F-distribution
(1,1)          0.0039876

```

There is a highly significant difference among the scales,  $P = .00399$ . Now we need to find out where the differences are using Bonferronized t-tests of the three pairwise differences among the scales. We enlarge  $c$  to include an A vs C contrast.

```

Cmd> c1 <- vconcat(c,vector(1,0,-1)');c1
(1,1)          1          -1          0          A vs B
(2,1)          0          1          -1          B vs C
(3,1)          1          0          -1          A vs C
Cmd> diffs <- vector(c1 %%% grandmean);diffs #diffs among grand means
(1)          -1.0952          -0.2381          -1.3333
Cmd> seDiffs <- sqrt(diag(c1 %%% vhat %%% c1'));seDiffs #Std Errors
(1)          0.3787          0.41505          0.35385
Cmd> tstats <- diffs/seDiffs; tstats # t-statistics
(1)          -2.8921          -0.57365          -3.7681
Cmd> 3*twotailt(tstats,fe) # two tail Bonferronized P-values
(1)          0.030395          1.7211          0.0046007

```

Scales B and C are not significantly different, but both are significantly different from scale A. This might be summarized by an “underline diagram” with the three scale means:

```

Cmd> grandmean
(1)          14.524          15.619          15.857
           A          B          C

```

## Profile Analysis Example

Now look at main effects between groups. This is based on the subject averages across all three scales.

```
Cmd> subjmeans <- describe(y',mean:T) # We work with subject means
Cmd> anova("subjmeans=groups") # univariate ANOVA
Model used is subjmeans=groups
WARNING: summaries are sequential
```

	DF	SS	MS
CONSTANT	1	4937.3	4937.3
groups	3	247.97	82.656
ERROR1	17	19.811	1.1654

```
Cmd> ms <- SS/DF; f <- ms[2]/ms[3]; f # ANOVA F-statistic
Cmd> cumF(f,DF[2],DF[3],upper:T) # P-value is extremely significant
(1) 8.0979e-10
```

Since a subject mean can be computed from a vector  $y$  of scores by the linear combination  $\mathbf{a}'y$  where  $\mathbf{a} = [1/3, 1/3, 1/3]'$ , you can also compute these SS's directly from MANOVA  $\mathbf{H}$  and  $\mathbf{E}$  as  $SS_h = \mathbf{a}'\mathbf{H}\mathbf{a}$  and  $SS_e = \mathbf{a}'\mathbf{E}\mathbf{a}$ . Alternatively, they are the averages of the  $3 \times 3 = 9$  elements of  $\mathbf{H}$  and  $\mathbf{E}$ .

```
Cmd> a <- rep(1,p)/p; vector(a' %*% h %*% a, a' %*% e %*% a)
      (1)      (2)
      247.97      19.811 Same as SSH and SSE in ANOVA
Cmd> # or from the averages of the elements of H and E
Cmd> describe(hconcat(vector(h),vector(e)),mean:T)
(1)      247.97      19.811
```

Now lets do a multiple comparison analysis of the 4 group means using Bonferroni two-sample t with standard errors computed from the MSE pooled across all groups.

```
Cmd> grp_aves <- vector(stats$mean %*% rep(1,p)/p,labels:"Class ")
Cmd> grp_aves
      Class 1      Class 2      Class 3      Class 4
      19.25      14.4      14      10
Cmd> # These are average across scales of group mean vectors
Cmd> diffs <- grp_aves - grp_aves'; diffs# all differences
(1,1)      0      4.85      5.25      9.25
(2,1)     -4.85      0      0.4      4.4
(3,1)     -5.25     -0.4      0      4
(4,1)     -9.25     -4.4     -4      0
Cmd> n <- tabs(,groups);n # get sample sizes
(1)      8      5      4      4
Cmd> mse <- ms[3] # pooled error mean square
Cmd> ses <- sqrt(mse*(1/n + 1/n')); ses # std errors
(1,1)      0.53976      0.61542      0.66107      0.66107
(2,1)      0.61542      0.68275      0.72416      0.72416
(3,1)      0.66107      0.72416      0.76333      0.76333
(4,1)      0.66107      0.72416      0.76333      0.76333
```

## Profile Analysis Example

```

Cmd> tstats <- diffs/ses;tstats
(1,1)      0      7.8808      7.9417      13.993
(2,1)     -7.8808      0      0.55236      6.076
(3,1)     -7.9417     -0.55236      0      5.2402
(4,1)    -13.993     -6.076     -5.2402      0

Cmd> ij <- hconcat(vector(1,1,1,2,2,3),vector(2,3,4,3,4,4));ij
(1,1)      1      2      Rows are i,j values to select
(2,1)      1      3      distinct t-statistics
(3,1)      1      4
(4,1)      2      3
(5,1)      2      4
(6,1)      3      4

Cmd> tstats <- tstats[ij]; tstats
(1)      7.8808      7.9417      13.993      0.55236      6.076
(6)      5.2402

Cmd> 6*twotailt(tstats,fe) # Bonferronized P-values
(1) 2.685e-06 2.4189e-06 5.5695e-10 3.5273 7.4102e-05
(6) 0.00039903

Cmd> 6*twotailt(tstats,fe) <= .05 # T means significant at 5% level
(1) T      T      T      F      T      T

Cmd> tcrit <- invstu(.025/6,fe,upper:T); tcrit # alternative
(1) 2.984

Cmd> abs(tstats) >= tcrit # same conclusion
(1) T      T      T      F      T      T

```

Groups 2 and 3 are not significantly different but all other differences are significant. You can summarize this with an "underline diagram".

```

Cmd> grp_aves[grade(grp_aves)] # sorted group means
      Class 4      Class 3      Class 2      Class 1
      10      14      14.4      19.25

```

Scheffé type comparisons for t from the F distribution are more conservative than t.

```

Cmd> scheffecrit<- sqrt(fh*invF(.05,fh,fe,upper:T)); scheffecrit
(1) 3.0968      > tcrit = 2.984

Cmd> abs(tstats) >= scheffecrit # same conclusion
(1) T      T      T      F      T      T

```

An alternative approach would be to consider this as analogous to a *split plot design* with subjects as whole plots and socioeconomic class and scale the whole plot and subplot "treatments", respectively.

To do this in MacAnova you need to turn y into a vector and create new factors, including a factor for subject within treatment. I chose to group all the values for each subject together.

```

Cmd> y1 <- vector(y') # make 3*N vector,"unraveling" by rows
Cmd> groups1 <- factor(vector(hconcat(groups,groups,groups)'))
Cmd> scales <- factor(rep(run(3),N)) # 1,2,3,1,2,3,..., 1,2,3
Cmd> tmp <- vector(run(8),run(5),run(4),run(4))
Cmd> subjects <- factor(vector(hconcat(tmp,tmp,tmp)'))

```

## Profile Analysis Example

```

Cmd> paste(subjects) #quick look at subjects
(1) "1 1 1 2 2 2 3 3 3 4 4 4 5 5 5 6 6 6 7 7 7 8 8 8 1 1 1 2 2 2
3 3 3 4 4 4 5 5 5 1 1 1 2 2 2 3 3 3 4 4 4 1 1 1 2 2 2 3 3 3 4 4 4"

Cmd> anova("y1=groups1+E(subjects.groups1)+scales+groups1.scales",\
          fstat:T)
Model used is y1=groups1+E(subjects.groups1)+scales+groups1.scales

```

	DF	SS	MS	F	P-value
CONSTANT	1	14812	14812	4236.74706	< 1e-08
groups1	3	743.9	247.97	70.92709	< 1e-08
ERROR1	17	59.433	3.4961	2.26557	0.020779
scales	2	21.238	10.619	<u>6.88147</u>	0.0030945
groups1.scales	6	18.962	3.1603	<u>2.04798</u>	0.085952
ERROR2	34	52.467	1.5431		

Note that the F-statistic for testing difference among the groups is the same as we derived before. Its validity depends on equality of variance matrices among the groups, or more accurately, equality of  $\mathbf{1}_p' \mathbf{\Sigma} \mathbf{1}_p = \sum_i \sum_j \sigma_{ij}$  among the groups.

The other two P-values are appropriate only if the assumptions for the analysis of variance are satisfied. In this context this means that the variances (diagonal elements  $\sigma_{ii}$  of  $\mathbf{\Sigma}$ ) are equal and all correlations  $\rho_{ij}$  are equal. Let's look to see how likely this is.

```

Cmd> s <- matrix(e/fe) # compute s from error matrix
Cmd> diag(s)
(1)      2.5176      1.8118      2.2529      Variances

Cmd> d <- dmat(1/sqrt(diag(s))); d %*% s %*% d
              (1)      (2)      (3)
(1)          1      0.30848    0.44952      Correlations
(2)      0.30848      1      0.11064
(3)      0.44952      0.11064      1

```

The assumptions don't seem too badly violated, especially in view of the small error degrees of freedom.

When the assumptions are *not* satisfied, but the groups have the same  $\mathbf{\Sigma}$ 's, the F-statistics still have a null distribution that is approximately F, but with modified degrees of freedom. Greenhouse and Geisser (*Psychometrika* **24** (1959) 95-112) have shown that the smallest the modified degrees of freedom can possibly be is  $1/(p-1)$  times the ANOVA degrees of freedom. Let's see what that does.

```

Cmd> mse <- SS[6]/DF[6]; mse
      ERROR2
      1.5431

Cmd> fscales <- (SS[4]/DF[4])/mse; fparallel <- (SS[5]/DF[5])/mse
Cmd> vector(fscales,fparallel) # same as in ANOVA output
(1)      6.8815      2.048

Cmd> df_min <- DF/(p-1); df_min [vector(4,5,6)] # minimum d.f.
      scalesgroups1.scales      ERROR2
      1          3          17

```



## Profile Analysis Example

```
Cmd> cumF(vector(fscales,fparallel),df_min[vector(4,5)],df_min[6],\
upper:T)
(1)      0.017802      0.14528 scales and interaction P-values
```

These last are conservative P-values since they are based on pessimistic estimates of the degrees of freedom. Both the unadjusted P-values 0.0030945 and 0.085952 and these conservative P-values lead to the same conclusion as the multivariate analysis, namely that interaction is not significantly different from 0 and among-scales differences are significant, at least using  $\alpha = .05$ .

What Greenhouse and Geisser actually showed was that you should adjust both numerator and denominator degrees of freedom by multiplying them by

$$\varepsilon \equiv \frac{p^2(\bar{\sigma}_d - \bar{\sigma})^2}{(p-1)(A_1 - 2A_2)},$$

where

$$A_1 = \sum_i \sum_j (\sigma_{ij} - \bar{\sigma}_{..})^2 \text{ and } A_2 = p \sum_j (\bar{\sigma}_{.j} - \bar{\sigma}_{..})^2.$$

Here  $\bar{\sigma}_d = \sum_i \sigma_{ii} / p = \text{tr} \mathbf{\Sigma} / p$  is the average of the diagonal elements of  $\mathbf{\Sigma}$ ,  $\bar{\sigma}_{..} = \sum_i \sum_j \sigma_{ij} / p^2 = \mathbf{1}_p' \mathbf{\Sigma} \mathbf{1}_p / p^2$  is the average of all the elements of  $\mathbf{\Sigma}$ , and  $\bar{\sigma}_{.j} = \sum_i \sigma_{ij} / p$  is the average of column  $j$  of  $\mathbf{\Sigma}$  (also average of row  $j$ , since  $\mathbf{\Sigma}' = \mathbf{\Sigma}$ ).

If you knew  $\mathbf{\Sigma}$  you could compute  $A_1$  as  $p^2 - 1$  times the “sample variance” of the elements of  $\mathbf{\Sigma}$ , and  $A_2$  as  $p(p-1)$  times the “sample variance” of the column means of  $\mathbf{\Sigma}$ . When you don't know  $\mathbf{\Sigma}$ , the best you can do is estimate  $\varepsilon$  by similar computations starting with  $\hat{\Sigma} = \mathbf{S}_p = \mathbf{E} / f_e$ , the pooled estimated variance matrix. You can get the “sample variance” of the elements of a matrix  $a$  by `describe(vector(a), var:T)`.

```
Cmd> sd <- trace(s)/p
Cmd> topeps <- p^2*(sd - sum(vector(s))/p^2)^2
Cmd> a1 <- (p*p-1)*describe(vector(s),var:T)
Cmd> a2 <- p*(p-1)*describe(describe(s,mean:T),var:T)
Cmd> bottomeps <- (p-1)*(a1 - 2*a2); epsilon <- topeps/bottomeps
Cmd> epsilon
(1)      0.96641 Not far from 1 -> little adjustment
Cmd> df <- epsilon*DF
Cmd> df[vector(4,5,6)] # readjusted d.f.
      scalesgroups1.scales      ERROR2
      1.9328      5.7984      32.858
Cmd> cumF(vector(fscales,fparallel),df[vector(4,5)],df[6],upper:T)
(1)      0.0034763      0.088938
```

Since  $\hat{\varepsilon}$  is so close to 1, the adjusted P-values are not very different from the unadjusted ones – 0.0030945 and 0.085952 – and give the same conclusion.