

## Stat 8053: Manova (rev November 7, 2011)

We begin by recalling some results from the univariate case. Suppose that  $Y \sim N_n(\mu, \sigma^2 I)$  for some unknown mean vector  $\mu$  and variance  $\sigma^2 > 0$ . By definition then,

$$(Y - \mu)'(Y - \mu)/\sigma^2 \sim \chi^2(n)$$

since a sum of  $n$  independent  $\chi^2(1)$  random variables has a  $\chi^2(n)$  distribution. If we don't adjust for the mean, then

$$Y'Y/\sigma^2 \sim \chi^2(n, \mu'\mu/\sigma^2)$$

a non-central chi-squared random variable with non-centrality parameter  $\mu'\mu/\sigma^2$ . All of linear model theory with one error term is based on restrictions of the vector  $\mu$ . It may be of some interest to note that  $Y'Y = Y'IY$  and that  $I$  is a projection matrix.

### Two-way Analysis of Variance

For an example, let's consider a two-factor problem in which factor A has 2 levels, factor B has two levels, and there are two observations per cell:

	$B = 1$	$B = 2$	$B = 3$
$A = 1$	$y_{111}, y_{112}$	$y_{121}, y_{122}$	$y_{131}, y_{132}$
$A = 2$	$y_{211}, y_{212}$	$y_{221}, y_{222}$	$y_{231}, y_{232}$

Let's collect  $Y$  into a  $12 \times 1$  vector  $y_{111}, y_{112}, \dots, y_{232}$ . A model is obtained by putting restrictions on  $\mu$ . Let's first fit the model

$$\mu = 1\beta_0$$

and so this "design" consists of a column of ones. Let  $J = 11'/1'1$  be the projection onto the column of 1's, so we have  $J = J'$ ,  $J^2 = J$  and since  $J1 = 1$ , the subspace spanned by  $J$  is the same as the subspace spanned by 1. If  $Z$  is any vector, then  $JZ = 11'Z/1'1 = 1(1'Z)/n = \bar{z}1$ , so any vector multiplied by  $J$  returns a vector that is just a constant times 1. The textbook defines  $H = I - J$ , which is also a projection matrix with  $HJ = JH = 0$ . Any product  $HZ$  returns a vector that is orthogonal to the vector of ones, which is accomplished by centering  $Z$  by subtracting  $\bar{y}1$ .

By Cochran's theorem, since  $J$  is a projection,  $Y'JY/\sigma^2 \sim \chi^2(1, \delta_1^2)$ . Now  $JY = 1(1'Y)/(1'1) = 1\bar{y}$ , and so  $Y'JY = n\bar{y}^2$ , as it should. Here  $\delta_1^2$  is the non-centrality parameter, which is equal to  $\mu'J\mu/\sigma^2 = n\beta_0^2/\sigma^2$ .

Let's move on to the one-way design for factor  $B$ , for which each level of  $B$  has its own mean. Define  $b_1$  to be dummy variable that is equal to one if the second subscript of the corresponding element of  $Y$  is equal to one and equal to zero otherwise. Define the dummy  $b_2$  and  $b_3$  similarly for levels 2 and 3 of  $B$ . Then we can write

$$\mu = b_1\tau_1 + b_2\tau_2 + b_3\tau_3$$

I'm going to create a vector  $B$  of 1s, 2s and 3s that corresponds to the level of  $B$  for the 12 responses, in *row order*, and from that create a design matrix whose columns are  $b_1, b_2, b_3$ .

```
> (B <- rep(gl(3, 2), 2))
```

```
[1] 1 1 2 2 3 3 1 1 2 2 3 3
Levels: 1 2 3
```

```
> (XB <- as.data.frame(model.matrix(~B - 1)))
```

```
      B1 B2 B3
1      1  0  0
2      1  0  0
3      0  1  0
4      0  1  0
5      0  0  1
6      0  0  1
7      1  0  0
8      1  0  0
9      0  1  0
10     0  1  0
11     0  0  1
12     0  0  1
```

By definition, each  $\tau_j$  is the mean value for observations in the  $j$ -th level of  $B$ . This parameterization of the  $B$  main-effect is not the standard one because it includes the overall mean since  $X_B 1 = 1$ , and so the column of one's is in the subspace spanned by the columns of  $X_B$ . It is usual to make the main effects be orthogonal to the overall mean. R does in two steps by first adding the vector 1 to the design matrix and deleting the dummy variable for the lowest level of  $B$ , and then when computing replacing the last two columns of  $B$  by two vectors that span the same subspace as that spanned by  $(b_2, b_2)$  by are orthogonal to 1. The model is

$$\mu = 1\beta_0 + b_2\tau_2 + b_3\tau_3$$

The meaning of  $\tau_2$  and  $\tau_3$  is different in these two representations of the model. One can choose any two linear combinations  $\sum c_j b_j$  and  $\sum d_j b_j$  such that the  $c_j$  are not all equal and the  $d_j$  are not all equal. This will change the meaning of parameters, but will not change subspaces. The design matrix created by R for this model is

```
> as.data.frame(model.matrix(~B))
```

```
 (Intercept) B2 B3
1             1  0  0
2             1  0  0
3             1  1  0
4             1  1  0
5             1  0  1
6             1  0  1
7             1  0  0
8             1  0  0
9             1  1  0
10            1  1  0
11            1  0  1
12            1  0  1
```

Before getting the orthogonal basis, we we continue by fitting the full model with interaction:

```
> (A <- gl(2, 6))
```

```
[1] 1 1 1 1 1 1 2 2 2 2 2 2
Levels: 1 2
```

```
> as.data.frame(X <- model.matrix(~1 + B + A + B:A))
```

```
      (Intercept) B2 B3 A2 B2:A2 B3:A2
1                1  0  0  0        0    0
2                1  0  0  0        0    0
3                1  1  0  0        0    0
4                1  1  0  0        0    0
5                1  0  1  0        0    0
6                1  0  1  0        0    0
7                1  0  0  1        0    0
8                1  0  0  1        0    0
9                1  1  0  1        1    0
10               1  1  0  1        1    0
11               1  0  1  1        0    1
12               1  0  1  1        0    1
```

The column of this matrix marked `Intercept` is for the subspace for the overall mean, the next two columns called `B2` and `B3` are for the main effect of  $B$ , the next one is for  $A$ , and 2 for the  $B:A$  interaction. The projections on the subspaces spanned by these four matrices are orthogonal, and lead to the usual analysis of variance.

The basis vectors (columns) corresponding to each of these subspaces are not orthogonal, and the simplest way to get the sums of squares, and apply Cochran's theorem, is to replace  $X$  by an orthogonal version. R, and many other programs, uses the QR factorization: There is an  $n \times p$  matrix  $Q$  and a  $p \times p$  upper triangular matrix  $R$  such that  $X = QR$ , and if  $X$  is of full rank  $Q = XR^{-1}$  and the columns of  $Q$  are just linear combinations of the columns of  $X$ .  $Q$  is useful because it has orthogonal columns, and (1)  $Q'Q = I$ ; (2)  $QQ'$  is the orthogonal projection on the columns of  $X$ . Also, if  $Q_j$  is the  $j$ th column of  $Q$ , then  $QQ' = \sum Q_j Q_j'$ , each of the  $Q_j Q_j'$  is a projection matrix, independent of the others, and  $Y'Q_j Q_j' Y = (Y'Q_j)^2 \sim \chi^2(1, (\mu'Q_j)^2/\sigma^2)$ . From this we get the usual analysis of variance. For example, the sum of squares for the  $B$  main effect is  $(Y'Q_2)^2 + (Y'Q_3)^2$ .

The function `qr` computes information needed to compute the QR-decomposition, and the function `qr.Q` applied to the output of `qr` returns the matrix  $Q$ . Similarly, `qr.R` returns  $R$ .

```
> zapsmall(Q <- qr.Q(qr(X)))
```

```
      [,1] [,2] [,3] [,4] [,5] [,6]
[1,] -0.2887 -0.2041 -0.3536 -0.2887 -0.2041  0.3536
[2,] -0.2887 -0.2041 -0.3536 -0.2887 -0.2041  0.3536
[3,] -0.2887  0.4082  0.0000 -0.2887  0.4082  0.0000
[4,] -0.2887  0.4082  0.0000 -0.2887  0.4082  0.0000
[5,] -0.2887 -0.2041  0.3536 -0.2887 -0.2041 -0.3536
[6,] -0.2887 -0.2041  0.3536 -0.2887 -0.2041 -0.3536
[7,] -0.2887 -0.2041 -0.3536  0.2887  0.2041 -0.3536
[8,] -0.2887 -0.2041 -0.3536  0.2887  0.2041 -0.3536
[9,] -0.2887  0.4082  0.0000  0.2887 -0.4082  0.0000
[10,] -0.2887  0.4082  0.0000  0.2887 -0.4082  0.0000
[11,] -0.2887 -0.2041  0.3536  0.2887  0.2041  0.3536
[12,] -0.2887 -0.2041  0.3536  0.2887  0.2041  0.3536
```

Finally, we need to discuss the error space. Suppose  $P_X$  is the projection on the column space of  $X$  (also the columns of  $Q$ ) with rank/dimension equal to  $q$ . Let  $P_X^\perp = I - P_X$ . It is straightforward to show that  $P_X^\perp$  is also a projection, that it is orthogonal to  $P_X$  and hence to the projections on each of the components of  $P_X$ , and  $Y'(I - P_X^\perp)Y \sim \sigma^2 \chi^2(n - q)$ , a central chi-square because  $P_X^\perp \mu = 0$ . Thus the unbiased estimate of  $\sigma^2$  is  $Y'(I - P_X^\perp)Y/(n - p)$ .

Here is the “by hand” computation of the sums of squares, and the same results using the `lm` function.

```
> set.seed(100)
> Y <- rnorm(12)
> YTQ <- t(Y) %*% Q
> (ss <- c(YTQ[2]^2 + YTQ[3]^2, YTQ[4]^2, YTQ[5]^2 + YTQ[6]^2,
+       sum(Y^2) - sum(YTQ^2)))

[1] 0.1464 0.2520 0.8199 1.6360

> anova(lm(Y ~ B + A + B:A))
```

Analysis of Variance Table

Response: Y

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
B	2	0.146	0.073	0.27	0.77
A	1	0.252	0.252	0.92	0.37
B:A	2	0.820	0.410	1.50	0.30
Residuals	6	1.636	0.273		

## One-way multivariate analysis of variance

The notation in this section differs slightly from the notation in the first section so be careful! The one-way Manova model is, for  $i = 1, \dots, g$ , and  $j = 1, \dots, n_i$ ,

$$X_{ij} \sim N_p(\mu_i, \Sigma)$$

with  $\mu_i = \mu_0 + \tau_i$ . Here  $\mu_0$  is a fixed  $p$  vector and the  $\tau_i$ , the group effects, are also fixed  $p$ -vectors, subject to a constraint to make the model full rank as in one-way anova. As in R, we will set  $\tau_1 = 0$ . The assumption of common  $\Sigma$  may be troubling, particularly if data are observational rather than from a randomized experiment.

If  $p = 1$  we get the one-way anova model. The estimates of the  $\mu_i$  are simply the  $\bar{x}_i$ , and if  $\bar{x} = \sum n_i \bar{x}_i / \sum n_i$ , the anova table is:

Source	df	SS
Groups	$g - 1$	$SSB = X'P_G X = \sum n_i (\bar{x}_i - \bar{x})^2$
Error	$\sum (n_i - 1)$	$SSW = X'(I - P_G - J)X = \sum \sum (x_{ij} - \bar{x}_i)^2$

Let  $G$  be an  $n \times (g - 1)$  matrix whose  $j$  column is the group identifier for group  $j$ ,  $j = 2, \dots, g$ . Then  $P_G$  is the projection on the columns of  $G$ , orthogonal to the columns of ones for the intercept, and  $P_G$  has rank  $g - 1$  and is orthogonal to  $J$  as defined in the text. The  $F$  test of all the  $\mu_i$  is

$$F = \frac{SSB/(g - 1)}{SSW/\sum(n_i - 1)}$$

which has a central  $F$  distribution under the null hypothesis and a non-central  $F$  distribution under the alternative.

When  $p > 1$ , the results are analogous. The mle of  $\mu_i$  is still  $\bar{x}_i$ ; that is, the individual elements of  $\mu_i$  are estimated as in univariate regression. The Manova table looks like this:

Source	df	SS
Groups	$g - 1$	$B = X'P_G X = \sum n_i(\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})'$
Error	$\sum(n_i - 1)$	$W = X'(I - J - P_G)X = \sum \sum (x_{ij} - \bar{x}_i)(x_{ij} - \bar{x}_i)'$

The estimate of  $\Sigma$  is  $W/(\sum(n_i - 1))$ .

The likelihood ratio test of all the  $\mu_i$  equal can be derived from the multivariate normal likelihood. It can be shown to be

$$\Lambda = \frac{|W|}{|W + B|} = \frac{1}{|I + W^{-1}B|}$$

where  $|A|$  is the determinant of  $A$ , and we reject for small values of  $\Lambda$ .  $\Lambda$  depends on the data only through the eigenvalues of the matrix  $W^{-1}B$ , which is a direct generalization of the  $F$  test in Anova. If  $\lambda_1 \geq \dots \lambda_p$  are the eigenvalues of  $W^{-1}B$ , then

$$\Lambda = \prod \frac{1}{1 + \lambda_i}$$

The distribution of  $\Lambda$  depends on the distribution of the eigenvalues. Here are the usual methods of obtaining  $p$ -values.

1. Asymptotic:  $-2 \log(\Lambda) \rightarrow \chi^2[p(g - 1)]$ .
2. Bartlett corrected with  $n = \sum n_i$ :

$$-(n - 1 + \frac{p + g}{2}) \log(\Lambda) \rightarrow \chi^2(p(g - 1))$$

3. The exact distribution is available in special cases with  $gp \leq 3$ .
4. Replace  $\Lambda$  by a different function of the  $\lambda$ s but still uses their asymptotic distribution to get  $p$ -values. The Lawley-Hotelling method uses  $\text{tr}(W^{-1}B) = \sum \lambda_i$ . A small-sample  $F$ -approximation is used to get  $p$ -values.
5. Pillai's trace criterion appears to be the most popular and is often the default in computer programs,  $\text{tr}(B(B + W)^{-1}) = \sum(\lambda_i/(1 + \lambda_i))$ . A small-sample  $F$ -approximation is used to get  $p$ -values.
6. Roy's largest root is the maximum eigenvalue of  $W(B + W)^{-1} = \lambda_1/(1 + \lambda_1)$ . It is equivalent to testing all possible univariate Anova's with response  $Y'a$  for all  $a$ , and then computing the  $p$ -value based on the maximum of all these tests. The distribution of this statistic is known and tables (or algorithms to compute tables) are available.

We use as an example data from Johnson and Wichern, *Applied Multivariate Statistical Analysis*, 6th Ed., p. 318. The data are concerned with a two-factor experiment done to study the properties of extruded plastic film. The two factors are rate of extrusion and amount of additive, both at two levels. There are three response variables,  $x_1 =$  tear resistance,  $x_2 =$  gloss, and  $x_3 =$  opacity. These three variables are on completely different scales and in different units.

```

> loc <- "http://www.stat.umn.edu/~sandy/courses/8053/Data/Wichern_data/T6-4.dat"
> data <- read.table(url(loc))
> colnames(data) <- c("Rate", "Additive", "X1", "X2", "X3")
> data$Rate <- factor(data$Rate)
> data$Additive <- factor(data$Additive)
> data$RateAdd <- factor(with(data, paste("R", Rate, "A", Additive,
+   sep = "")))
> data

```

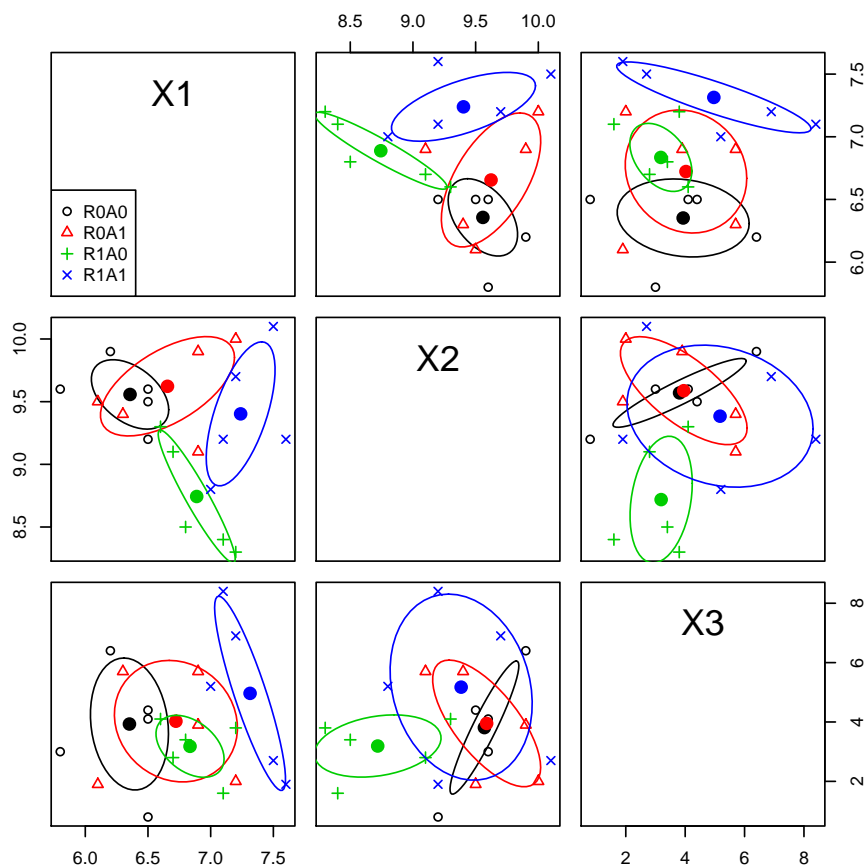
	Rate	Additive	X1	X2	X3	RateAdd
1	0	0	6.5	9.5	4.4	ROAO
2	0	0	6.2	9.9	6.4	ROAO
3	0	0	5.8	9.6	3.0	ROAO
4	0	0	6.5	9.6	4.1	ROAO
5	0	0	6.5	9.2	0.8	ROAO
6	0	1	6.9	9.1	5.7	ROA1
7	0	1	7.2	10.0	2.0	ROA1
8	0	1	6.9	9.9	3.9	ROA1
9	0	1	6.1	9.5	1.9	ROA1
10	0	1	6.3	9.4	5.7	ROA1
11	1	0	6.7	9.1	2.8	R1AO
12	1	0	6.6	9.3	4.1	R1AO
13	1	0	7.2	8.3	3.8	R1AO
14	1	0	7.1	8.4	1.6	R1AO
15	1	0	6.8	8.5	3.4	R1AO
16	1	1	7.1	9.2	8.4	R1A1
17	1	1	7.0	8.8	5.2	R1A1
18	1	1	7.2	9.7	6.9	R1A1
19	1	1	7.5	10.1	2.7	R1A1
20	1	1	7.6	9.2	1.9	R1A1

```

> library(car)
> print(scatterplotMatrix(~X1 + X2 + X3 | RateAdd, data = data,
+   smooth = FALSE, reg.line = FALSE, ellipse = TRUE, levels = c(0.5),
+   by.groups = TRUE, diagonal = "none"))

```

NULL



Let's start by ignoring the factors, and treat the last three columns as  $X$  with  $N(\mu, \Sigma)$  rows. Here is the one-sample test of  $\mu = 0$ , done by 'hand'.

```
> xbar <- colMeans(data[, 3:5])
> n <- dim(data)[1]
> S <- ((n - 1)/n) * var(data[, 3:5])
> (F0 <- ((n - 3)/3) * t(xbar) %*% solve(S) %*% (xbar))

      [,1]
[1,] 3783
```

You can get `lm` to compute the same test for you:

```
> (anova(m0 <- lm(cbind(X1, X2, X3) ~ 1, data)))
```

Analysis of Variance Table

	Df	Pillai	approx F	num Df	den Df	Pr(>F)
(Intercept)	1	0.999	3783	3	17	<2e-16
Residuals	19					

The computation by hand matches the the  $F$  computed from the Pillai statistic because there is only one eigenvalue for the test of  $\mu = 0$ . A slight modification can test for example  $\mu = \mu_0 = c(7, 9, 4)$ :

```
> mu0 <- c(7, 9, 4)
> anova(m1 <- update(m0, I(cbind(X1, X2, X3) - outer(rep(1, n),
+ mu0)) ~ .))
```

Analysis of Variance Table

	Df	Pillai	approx F	num Df	den Df	Pr(>F)
(Intercept)	1	0.348	3.02	3	17	0.059
Residuals	19					

The value I chose for  $\mu_0$  is `round(xbar)`, and even so the null hypothesis is almost rejected.

Let's fit a one-way Manova with four levels created all possible combinations of the factors:

```
> m1 <- lm(cbind(X1, X2, X3) ~ RateAdd, data)
> print(anova(m1, test = "Wilks"), digits = 8)
```

Analysis of Variance Table

	Df	Wilks	approx F	num Df	den Df	Pr(>F)
(Intercept)	1	0.00078358	5950.9058	3	14.000000	< 2.22e-16
RateAdd	3	0.17801870	3.9252	9	34.222927	0.0016629
Residuals	16					

## Two-way Manova

Just as one-way Anova generalizes easily to multiple factors and designs so does Manova. The multivariate linear model assuming common  $\Sigma$  can be fit using the `lm`:

```
> m2 <- lm(cbind(X1, X2, X3) ~ Rate * Additive, data)
> summary(m2)
```

Response X1 :

Call:

```
lm(formula = X1 ~ Rate * Additive, data = data)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.580	-0.205	0.060	0.220	0.520

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	6.300	0.148	42.43	<2e-16
Rate1	0.580	0.210	2.76	0.014
Additive1	0.380	0.210	1.81	0.089
Rate1:Additive1	0.020	0.297	0.07	0.947

Residual standard error: 0.332 on 16 degrees of freedom

Multiple R-squared: 0.586, Adjusted R-squared: 0.509

F-statistic: 7.56 on 3 and 16 DF, p-value: 0.00227

Response X2 :

Call:

```
lm(formula = X2 ~ Rate * Additive, data = data)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.600	-0.245	-0.070	0.325	0.700

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	9.560	0.181	52.75	<2e-16
Rate1	-0.840	0.256	-3.28	0.0047
Additive1	0.020	0.256	0.08	0.9388
Rate1:Additive1	0.660	0.362	1.82	0.0874

Residual standard error: 0.405 on 16 degrees of freedom

Multiple R-squared: 0.483, Adjusted R-squared: 0.386

F-statistic: 4.99 on 3 and 16 DF, p-value: 0.0125

Response X3 :

Call:

```
lm(formula = X3 ~ Rate * Additive, data = data)
```

Residuals:

Min	1Q	Median	3Q	Max
-3.12	-1.61	0.22	1.18	3.38

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	3.740	0.901	4.15	0.00075
Rate1	-0.600	1.274	-0.47	0.64403
Additive1	0.100	1.274	0.08	0.93841
Rate1:Additive1	1.780	1.802	0.99	0.33789

Residual standard error: 2.01 on 16 degrees of freedom

Multiple R-squared: 0.125, Adjusted R-squared: -0.039

F-statistic: 0.762 on 3 and 16 DF, p-value: 0.531

```
> print(anova(m2, test = "Wilks"), digits = 8)
```

Analysis of Variance Table

	Df	Wilks	approx F	num Df	den Df	Pr(>F)
(Intercept)	1	0.00078358	5950.9058	3	14	< 2e-16
Rate	1	0.38185838	7.5543	3	14	0.003034

```

Additive      1 0.52303490    4.2556    3    14 0.024745
Rate:Additive 1 0.77710576    1.3385    3    14 0.301782
Residuals    16

```

```
> print(anova(m2), digits = 8)
```

Analysis of Variance Table

```

              Df      Pillai approx F num Df den Df  Pr(>F)
(Intercept)   1 0.99921642 5950.9058     3    14 < 2e-16
Rate          1 0.61814162   7.5543     3    14 0.003034
Additive      1 0.47696510   4.2556     3    14 0.024745
Rate:Additive 1 0.22289424   1.3385     3    14 0.301782
Residuals    16

```

There is also an equivalent to the aov function in R:

```

> m3 <- manova(cbind(X1, X2, X3) ~ Rate * Additive, data)
> summary(m3)

```

```

              Df Pillai approx F num Df den Df Pr(>F)
Rate          1  0.618     7.55     3    14  0.003
Additive      1  0.477     4.26     3    14  0.025
Rate:Additive 1  0.223     1.34     3    14  0.302
Residuals    16

```

As with standard analysis of variance, there is an issue of how to compute tests if data are unbalanced. The results apparently parallel univariate anova. To get the correct order of fitting, use the `Anova` function in the `car` package. We delete two observations to make the data unbalanced.

```

> m3 <- update(m2, subset = -c(7, 17))
> Anova(m3)

```

Type II MANOVA Tests: Pillai test statistic

```

              Df test stat approx F num Df den Df Pr(>F)
Rate          1   0.632     6.86     3    12 0.0061
Additive      1   0.560     5.08     3    12 0.0168
Rate:Additive 1   0.411     2.80     3    12 0.0856

```

## Multivariate Linear Regression

The model is

$$\mathbf{Y} = \mathbf{XB} + \mathbf{e}$$

where  $\mathbf{Y}$  is  $n \times m$ ,  $\mathbf{X}$  is  $n \times p$ ,  $\mathbf{B}$  is  $p \times m$  and rows of  $\mathbf{Y}$  have  $N_p(x'_i \mathbf{B}, \Sigma)$  distributions. Conceptually this causes no new issues. We use the `sleep1` data from `alr3`. The responses are hours of slow-wave sleep `SWS`, hours of paradoxical sleep `PS`, and the predictors are body measurements, gestation period `GP`, and an index of predation with five levels. Units are mammal species.

```

> library(alr3)
> sleep1$D <- factor(sleep1$D)
> sleep1 <- na.omit(sleep1)
> s1 <- lm(cbind(SWS, PS) ~ log(BodyWt) + log(BrainWt) + log(GP) +
+       D, sleep1)
> summary(s1)

```

Response SWS :

Call:

```
lm(formula = SWS ~ log(BodyWt) + log(BrainWt) + log(GP) + D,
    data = sleep1)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-6.649	-1.211	0.409	1.357	5.549

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	14.35005	2.42321	5.92	1.1e-06
log(BodyWt)	-0.57594	0.47824	-1.20	0.2368
log(BrainWt)	0.00701	0.63332	0.01	0.9912
log(GP)	-0.77421	0.64896	-1.19	0.2411
D2	-1.31503	1.17743	-1.12	0.2719
D3	-3.50027	1.32382	-2.64	0.0123
D4	-1.72507	1.24434	-1.39	0.1747
D5	-4.29014	1.55728	-2.75	0.0094

Residual standard error: 2.65 on 34 degrees of freedom

Multiple R-squared: 0.605, Adjusted R-squared: 0.524

F-statistic: 7.44 on 7 and 34 DF, p-value: 1.99e-05

Response PS :

Call:

```
lm(formula = PS ~ log(BodyWt) + log(BrainWt) + log(GP) + D, data = sleep1)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-2.6444	-0.4605	0.0548	0.4049	1.8863

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	6.952	0.827	8.41	8.1e-10
log(BodyWt)	0.453	0.163	2.78	0.0088
log(BrainWt)	-0.436	0.216	-2.02	0.0514
log(GP)	-0.714	0.221	-3.23	0.0028
D2	-1.022	0.402	-2.54	0.0157

D3	-1.415	0.452	-3.13	0.0035
D4	-1.466	0.424	-3.46	0.0015
D5	-2.392	0.531	-4.50	7.5e-05

Residual standard error: 0.904 on 34 degrees of freedom  
Multiple R-squared: 0.649, Adjusted R-squared: 0.577  
F-statistic: 8.98 on 7 and 34 DF, p-value: 3.11e-06

```
> s0 <- update(s1, ~1)
> anova(s0, s1)
```

Analysis of Variance Table

Model 1: cbind(SWS, PS) ~ 1

Model 2: cbind(SWS, PS) ~ log(BodyWt) + log(BrainWt) + log(GP) + D

	Res.Df	Df	Gen.var.	Pillai	approx	F	num	Df	den	Df	Pr(>F)
1	41		4.56								
2	34	-7	2.32	1.11	6.03	14	68				1.6e-07

```
> Anova(s1)
```

Type II MANOVA Tests: Pillai test statistic

	Df	test	stat	approx	F	num	Df	den	Df	Pr(>F)
log(BodyWt)	1	0.252	5.56	2	33	0.0083				
log(BrainWt)	1	0.113	2.11	2	33	0.1377				
log(GP)	1	0.237	5.14	2	33	0.0114				
D	4	0.524	3.02	8	68	0.0059				