Stat 8311 – Estimating treatment means, unbalanced data

```
> searle <- data.frame(soil = rep(c("s1", "s2"), c(7, 8)),
+     var = c("v1", "v2", "v3")[c(1, 1, 1, 2, 2, 3, 3, 1, 1,
+         1, 1, 2, 3, 3, 3)], y = c(6, 10, 11, 13, 15, 14,
+         22, 12, 15, 19, 18, 31, 18, 9, 12))
```

Here are the observed row means, column means and cell means:

```
> mean(searle$y)

[1] 15

> tapply(searle$y, searle$soil, mean)

   s1    s2
13.00 16.75

> tapply(searle$y, searle$var, mean)

      v1       v2       v3
13.00000 19.66667 15.00000

> xtabs(y ~ soil + var, data = searle)/xtabs(~soil + var, data = searle)

    var
soil v1 v2 v3
  s1  9 14 18
  s2 16 31 13
```

Next, fit using two orders. I'll use the SAS contrasts, although *the result is invariant under reparameterization because only projections are used,* so the results can be compared to SAS.

```
> opt <- options(contrasts = c("contr.SAS", "contr.poly"))
> m1 <- aov(y ~ soil * var, data = searle)
> m2 <- update(m1, ~var * soil)
```

Fitted means are evidentally based on the projections onto various subspaces. Consider first the projections for model `m1`.

```
> (p1 <- proj(m1)[, ])

   (Intercept)  soil        var  soil:var   Residuals
1           15 -2.00 -2.7659574 -1.234043 -3.0000e+00
2           15 -2.00 -2.7659574 -1.234043  1.0000e+00
3           15 -2.00 -2.7659574 -1.234043  2.0000e+00
4           15 -2.00  5.0531915 -4.053191 -1.0000e-00
5           15 -2.00  5.0531915 -4.053191  1.0000e-00
6           15 -2.00 -0.9042553  5.904255 -4.0000e+00
7           15 -2.00 -0.9042553  5.904255  4.0000e+00
8           15  1.75 -1.6755319  0.925532 -4.0000e+00
9           15  1.75 -1.6755319  0.925532 -1.0000e-00
10          15  1.75 -1.6755319  0.925532  3.0000e+00
```

```
11          15  1.75 -1.6755319  0.925532  2.0000e+00
12          15  1.75  6.1436170  8.106383 -4.1281e-16
13          15  1.75  0.1861702 -3.936170  5.0000e+00
14          15  1.75  0.1861702 -3.936170 -4.0000e+00
15          15  1.75  0.1861702 -3.936170 -1.0000e+00
```

In examining `p1`, the first column, the projection on the column of ones, we get one value for each observation, always equal to $\bar{y}_{+++}$. The second column is the projection of y on the part of the `soil` space that is orthogonal to the column of ones, relative to the correct inner product. Since `soil` has two levels, this is the projection on a one-dimensional subspace, and the projection is equal to -2 for observations with `soil` = `s1`, and 1.75 for `soil` = `s2`. It is therefore reasonable to for the means for soils to be $15 - 2 = 13$ for `s1` and $15 + 1.75 = 16.75$ for `s2`.

Next consider the column for `var`. This is the projection on a two dimensional subspace for the three varieties, after adjusting for the overall mean and for *soil*. *The number of unique values for this projection is not equal to three, but rather there are six unique values, apparently corresponding to the six cells in the table.* Thus it is not clear how one would define a mean for each of the three varieties. What R appears to do is *compute a weighted average of the cell means:*

```
> tapply(p1[, 1] + p1[, 3], searle$var, mean)

      v1       v2       v3
12.85714 20.41667 14.75000
```

Finally the (`soil`,`var`) means are simply the cell means

```
> xtabs(apply(p1[, -5], 1, sum) ~ soil + var, data = searle)/xtabs(~soil +
+     var, data = searle)

    var
soil v1 v2 v3
  s1  9 14 18
  s2 16 31 13

> print(model.tables(m1, type = "mean"), digits = 6)

Tables of means
Grand mean

15

 soil
    s1    s2
    13 16.75
rep  7  8.00

 var
         v1      v2    v3
    12.8571 20.4167 14.75
rep  7.0000  3.0000  5.00
```

```
 soil:var
     var
soil  v1 v2 v3
  s1   9 14 18
  rep  3  2  2
  s2  16 31 13
  rep  4  1  3
```

```
> proj(m2 <- update(m1, ~var * soil))[, ]
```

```
   (Intercept)           var      soil  var:soil     Residuals
1           15 -2.000000e+00 -2.765957 -1.234043 -3.000000e+00
2           15 -2.000000e+00 -2.765957 -1.234043  1.000000e+00
3           15 -2.000000e+00 -2.765957 -1.234043  2.000000e+00
4           15  4.666667e+00 -1.613475 -4.053191 -1.000000e+00
5           15  4.666667e+00 -1.613475 -4.053191  1.000000e+00
6           15  8.881784e-16 -2.904255  5.904255 -4.000000e+00
7           15  8.881784e-16 -2.904255  5.904255  4.000000e+00
8           15 -2.000000e+00  2.074468  0.925532 -4.000000e+00
9           15 -2.000000e+00  2.074468  0.925532 -1.000000e-00
10          15 -2.000000e+00  2.074468  0.925532  3.000000e+00
11          15 -2.000000e+00  2.074468  0.925532  2.000000e+00
12          15  4.666667e+00  3.226950  8.106383 -1.761829e-16
13          15  8.881784e-16  1.936170 -3.936170  5.000000e+00
14          15  8.881784e-16  1.936170 -3.936170 -4.000000e+00
15          15  8.881784e-16  1.936170 -3.936170 -1.000000e+00
```

```
> model.tables(m2, type = "mean")
```

```
Tables of means
Grand mean

15

 var
    v1    v2 v3
    13 19.67 15
rep  7  3.00  5

 soil
       s1    s2
    12.52 17.17
rep  7.00  8.00

 var:soil
     soil
var   s1 s2
  v1   9 16
  rep  3  4
  v2  14 31
```

```
 rep  2  1
 v3  18 13
 rep  2  3
```

```
options nocenter ls=75;
proc glm data=work.searle;
 class var soil;
 model days = var soil var*soil/ss1 ss2 ss3 solution;
 lsmeans var soil var*soil;
 run;
```

```
Dependent Variable: days
                                    Sum of
 Source                     DF       Squares   Mean Square  F Value  Pr > F
 Model                       5   400.0000000    80.0000000     6.00  0.0103
 Error                       9   120.0000000    13.3333333
 Corrected Total            14   520.0000000
           R-Square     Coeff Var      Root MSE     days Mean
           0.769231      24.34322      3.651484      15.00000

 Source                     DF     Type I SS   Mean Square  F Value  Pr > F
 var                         2    93.3333333    46.6666667     3.50  0.0751
 soil                        1    83.9007092    83.9007092     6.29  0.0334
 var*soil                    2   222.7659574   111.3829787     8.35  0.0089

 Source                     DF    Type II SS   Mean Square  F Value  Pr > F
 var                         2   124.7340426    62.3670213     4.68  0.0405
 soil                        1    83.9007092    83.9007092     6.29  0.0334
 var*soil                    2   222.7659574   111.3829787     8.35  0.0089

 Source                     DF   Type III SS   Mean Square  F Value  Pr > F
 var                         2   192.1276596    96.0638298     7.20  0.0135
 soil                        1   123.7714286   123.7714286     9.28  0.0139
 var*soil                    2   222.7659574   111.3829787     8.35  0.0089

Parameter estimates
Parameter             Estimate             Error    t Value    Pr > |t|

Intercept          13.00000000 B      2.10818511       6.17      0.0002
var       1         3.00000000 B      2.78886676       1.08      0.3100
var       2        18.00000000 B      4.21637021       4.27      0.0021
var       3         0.00000000 B       .                .         .
soil      1         5.00000000 B      3.33333333       1.50      0.1679
soil      2         0.00000000 B       .                .         .
var*soil  1 1     -12.00000000 B      4.34613494      -2.76      0.0221
var*soil  1 2       0.00000000 B       .                .

The GLM Procedure  Least Squares Means
var     days LSMEAN
1        12.5000000
```

```
2        22.5000000
3        15.5000000

soil     days LSMEAN
1         13.6666667
2         20.0000000

var    soil     days LSMEAN
1      1          9.0000000
1      2         16.0000000
2      1         14.0000000
2      2         31.0000000
3      1         18.0000000
3      2         13.0000000
     .                .
```

Disconnected data

The Big Idea with disconnected data is that the dimensions of subspaces may change depending on the order of fitting.

```
> xtabs(y ~ row + col, data = d1 <- data.frame(row = factor(c(1:4,
+     4)), col = factor(c(1:4, 3)), y = 1:5))

   col
row 1 2 3 4
  1 1 0 0 0
  2 0 2 0 0
  3 0 0 3 0
  4 0 0 5 4

> (a1 <- aov(y ~ row + col, data = d1))

Call:
   aov(formula = y ~ row + col, data = d1)

Terms:
                row col
Sum of Squares  9.5 0.5
Deg. of Freedom   3   1

2 out of 7 effects not estimable
Estimated effects may be unbalanced

> (a2 <- update(a1, ~col + row))

Call:
   aov(formula = y ~ col + row, data = d1)

Terms:
```

```
                col row
Sum of Squares    8   2
Deg. of Freedom   3   1

2 out of 7 effects not estimable
Estimated effects may be unbalanced

> model.tables(a1, type = "m")

Tables of means
Grand mean

3

 row
    1 2 3   4
    1 2 3 4.5
rep 1 1 1 2.0

 col
    1 2    3   4
    3 3 3.25 2.5
rep 1 1 2.00 1.0
```

The `Anova` command in R fails with empty cells because the dimensions of subspaces change, depending on the order of fitting. This could be avoided if `Anova` refit models rather than using the Wald approach to testing.

SAS recognizes that the disconnected data has row and column effects that are not estimable.