Least Squares Estimation

Peter Claussen

July 25, 2017

Contents

We outline the steps of least squares estimation in ARM.

Example Data and Analysis of Variance

Consider an example randomized complete block design with missing data, from (Giesbrecht and Gumpertz 2004).

Table 1: Simulated Data for RCBD with Four Blocks, Six Treatments and Two Missing Observations

Block	1	2	3	4	5	6
1	-	2.66	2.80	3.43	5.25	6.47
2	-	6.41	7.04	6.09	5.69	8.43
3	4.93	6.07	6.19	3.26	4.99	5.95
4	3.13	4.48	5.22	6.71	6.34	7.35

```r
data(Geisbrecht)
Table5.7 <- subset(Table5.7,!is.na(Table5.7$obs))
head(Table5.7)
```

```
##   plot rep col trt  obs
## 1  101   1   1   4 3.43
## 2  102   1   2   5 5.25
## 3  103   1   3   6 6.47
## 4  104   1   4   3 2.80
## 6  106   1   6   2 2.66
## 7  201   2   1   6 8.43
```

```r
tbl5.7.lm <- lm(obs ~ trt + rep, data=Table5.7)
tbl5.7.aov <- aov(tbl5.7.lm)
```

```r
TrialName="Trial"
```

Analysis of Variance, Simulated Data for RCBD with Four Blocks, Six Treatments and Two Missing Observations

```r
summary(tbl5.7.aov)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)  
## trt          5  16.88   3.376   2.568 0.0793 .
## rep          3  17.57   5.858   4.456 0.0232 *
## Residuals   13  17.09   1.315                 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ARM R AOV

```r
res.mar <- analyze.ARM.rcb(d=Table5.7,response="obs", TrtName="trt", RepName="rep")
make.table(res.mar$AovVar)
```

```
##           Df       SS       MS     Fval        prF
## Treatment  5 16.87941 3.375881 2.567911 0.07927654
## Rep        3 17.57459 5.858196 4.456118 0.02316320
## Error     13 17.09034 1.314641       NA         NA
```

AOV Means Table, Simulated Data for RCBD with Four Blocks, Six Treatments and Two Missing Observations

```r
mns.tbl <- data.frame(Arith=res.mar$raw_means,LSMean=res.mar$ls_means$means)
mns.tbl
```

```
##    Arith LSMean
## 1 4.0300 3.9155
## 2 4.9050 4.9050
## 3 5.3125 5.3125
## 4 4.8725 4.8725
## 5 5.5675 5.5675
## 6 7.0500 7.0500
```

How does ARM compute adjusted means for this data?

Means estimates for treatments 1-6, $\hat{\mu} = \hat{\mu}_1 \ldots \hat{\mu}_6$, are calculated as a linear combination of the least squares estimate of parameters associated with treatments and blocks, such that

$$
\begin{aligned}
\hat{\mu} &= L\hat{\beta} \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ 1 & 1 & 0 & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ 1 & 0 & 1 & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ 1 & 0 & 0 & 1 & 0 & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ 1 & 0 & 0 & 0 & 1 & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ 1 & 0 & 0 & 0 & 0 & 1 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix} \times \begin{bmatrix} 2.4960 \\ 0.9895 \\ 1.3970 \\ 0.9570 \\ 1.6520 \\ 3.1345 \\ 2.6100 \\ 1.3807 \\ 1.6873 \end{bmatrix}^t \\
&= \begin{bmatrix} 3.9155 \\ 4.9050 \\ 5.3125 \\ 4.8725 \\ 5.5675 \\ 7.0500 \end{bmatrix}
\end{aligned}
$$

LS linear combinations

These values are returned from the ARMR functions as members of an object:

```
res.mar$ls_means$L
```

```
##     2 3 4 5 6
## 1 1 0 0 0 0 0 0.25 0.25 0.25
## 2 1 1 0 0 0 0 0.25 0.25 0.25
## 3 1 0 1 0 0 0 0.25 0.25 0.25
## 4 1 0 0 1 0 0 0.25 0.25 0.25
## 5 1 0 0 0 1 0 0.25 0.25 0.25
## 6 1 0 0 0 0 1 0.25 0.25 0.25
```

Table:

β

```
res.mar$ls_means$beta
```

```
## (Intercept)        trt2        trt3        trt4        trt5        trt6 
##    2.496000    0.989500    1.397000    0.957000    1.652000    3.134500 
##        rep2        rep3        rep4 
##    2.610000    1.380667    1.687333
```

Variance-Covariance

```
res.mar$varCov
```

```
##             [,1]        [,2]        [,3]        [,4]        [,5]
## [1,]  0.5477672 -0.10955344 -0.10955344 -0.10955344 -0.10955344
## [2,] -0.1095534  0.28483894 -0.04382138 -0.04382138 -0.04382138
## [3,] -0.1095534 -0.04382138  0.28483894 -0.04382138 -0.04382138
## [4,] -0.1095534 -0.04382138 -0.04382138  0.28483894 -0.04382138
## [5,] -0.1095534 -0.04382138 -0.04382138 -0.04382138  0.28483894
## [6,] -0.1095534 -0.04382138 -0.04382138 -0.04382138 -0.04382138
##              [,6]
## [1,] -0.10955344
## [2,] -0.04382138
## [3,] -0.04382138
## [4,] -0.04382138
## [5,] -0.04382138
## [6,]  0.28483894
```

Information Matrix

```
res.mar$A
```

```
##            1          2          3          4          5          6
## 1  1.6666667 -0.3333333 -0.3333333 -0.3333333 -0.3333333 -0.3333333
## 2 -0.3333333  3.2666667 -0.7333333 -0.7333333 -0.7333333 -0.7333333
## 3 -0.3333333 -0.7333333  3.2666667 -0.7333333 -0.7333333 -0.7333333
## 4 -0.3333333 -0.7333333 -0.7333333  3.2666667 -0.7333333 -0.7333333
## 5 -0.3333333 -0.7333333 -0.7333333 -0.7333333  3.2666667 -0.7333333
## 6 -0.3333333 -0.7333333 -0.7333333 -0.7333333 -0.7333333  3.2666667
```

Concurrence

```
res.mar$concurrence
```

```
##   1 2 3 4 5 6
## 1 2 0 0 0 0 0
## 2 0 4 0 0 0 0
## 3 0 0 4 0 0 0
## 4 0 0 0 4 0 0
## 5 0 0 0 0 4 0
## 6 0 0 0 0 0 4
```

Parameters describing this experiment (i.e. treatment and block effects), $\hat{\beta}$ are estimated to minimize the residual SS, given by

$$SS_{Res} = (Y - X\hat{\beta})^T(Y - X\hat{\beta})$$

where Y is the set of observations in table 1 and X represents the details of the experimental design for this data. Specifically,

$$Y = \begin{bmatrix} 4.93 \\ 3.13 \\ 2.66 \\ 6.41 \\ 6.07 \\ \cdots \\ 7.35 \end{bmatrix}, X = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

For both X and L, column 1 represents an intercept, columns 2-6 indicate treatment number and columns 7-9 indicate replicates. More generally, both can be thought of as a combination of three matrices; one for grand mean, one representing treatment structure and one for blocking, for example,

$$X = X_0|X_1|X_2$$

The difference is that each row in X represents a unique value in Y, while rows in L are unique only for treatment parameters and average over the remaining parameters of $\hat{\beta}$. In this case, since $\hat{\beta}$ is estimated using ordinary least squares methods, the parameters estimated by $\hat{\mu}$ are commonly called least squares (treatment) means.

How does ARM find $\hat{\beta}$?

If X were a square matrix, then an exact solution β for can be found by inverting and multiplying,

$$\beta = X^{-1}Y$$

However, for most real data, there are more rows than columns in X and no unique inverse can be found, thus no exact solution for β. However, an estimate $\hat{\beta}$ can be obtained, and the preferred estimate minimizes residual SS. The result is the ordinary least squares estimate of β.

ARM calls an external mathematical library (CLAPACK) function (dgelsd) to find this solution using singular-value decomposition. ARM ST calls the R lm function, which uses QR factorization. Both methods produce identical results (within rounding error) in most cases.

Why are there only 5 treatment columns in X (and only 3 replicate columns)?

Consider a matrix with columns for grand mean, each treatment effect and each replicate effect:

$$X^* = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

We can see that the first column is redundant; it is simply the sum of the next 6 columns. Further, one of the last four columns is redundant in that it can be expressed as a linear combination of the other three $X^*_{i,8} = 1 - \sum(X^*_{i,9} \dots X^*_{i,11})$. This complicates finding solutions for X, so we remove redundant columns by using an effects encoding. ARM use an encoding that represents to the lowest level of each effect as the intercept (that is, column one represents treatment 1 in replicate 1), removing the corresponding coefficients from β. This convention is typically the default used in R, while SAS encodes the intercept as the highest-numbered level of each effect. Using this coding scheme, we see that the number of columns required for treatments (5) and replicates (3) are equal to the number of degrees of freedom for treatments and replicates, respectively, in Table 2.

How does ARM perform mean separations?

Consider first comparisons using LSD. ARM computes an LSD table, with entries for each treatment, such that the LSD for comparing treatments i and j are given by

$$LSD_{i,j} = t_{\alpha/2,\nu}\sqrt{V_{i,i} + V_{j,j} - V_{i,j} - V_{j,i}}$$

where

$$\begin{aligned} V &= A^{-1}\sigma^2 \\ &= \begin{bmatrix} 0.55477 & -0.10955 & \cdots & -0.10955 \\ -0.10955 & 0.28484 & \cdots & -0.04382 \\ \vdots & \vdots & \ddots & \vdots \\ -0.10955 & -0.04382 & & 0.28484 \end{bmatrix} \end{aligned}$$

The information matrix A is given by

$$\begin{aligned} A = & \qquad X_1 \times X_1^t - N X_2 \times {X_2^t}^{-1} N^t \\ = & \begin{bmatrix} 1.66667 & -0.33333 & \cdots & -0.33333 \\ -0.33333 & 3.26666 & \cdots & -0.73333 \\ \vdots & \vdots & \ddots & \vdots \\ -0.33333 & -0.73333 & \cdots & 3.26666 \end{bmatrix} \end{aligned}$$

Where N describes the coincidence of treatments (X_1) with blocks (X_2), $N = X_1^t X_2$.

```
TreatmentX <- incidence.matrix(Table5.7,effect="trt")
RepNoZ <- incidence.matrix(Table5.7,effect="rep")

N <- t(TreatmentX) %*% RepNoZ
head(N)
```

```
##   1 2 3 4
## 1 0 0 1 1
## 2 1 1 1 1
## 3 1 1 1 1
## 4 1 1 1 1
## 5 1 1 1 1
## 6 1 1 1 1
```

John and Williams (John and Williams 1995) use the notation t^δ to denote the replication matrix. In the balanced case, this matrix gives the replications of t.

```
t_delta <- t(TreatmentX) %*% TreatmentX
head(t_delta)
```

```
##   1 2 3 4 5 6
## 1 2 0 0 0 0 0
## 2 0 4 0 0 0 0
## 3 0 0 4 0 0 0
## 4 0 0 0 4 0 0
## 5 0 0 0 0 4 0
## 6 0 0 0 0 0 4
```

$v^{-\delta}$ is the inverse of v^{δ}. This can be thought of as determining the amount of information contained in replicates. This will be subtracted from the information represented by the total number of observations of each treatment, t^{δ}. If the data are balanced, we could simply divide, $N \times N'/v$, but when data are unbalanced this division is biased.

```
v_delta <- t(RepNoZ) %*% RepNoZ
v_delta_inv <- solve(v_delta)
v_delta_inv
```

```
##     1   2         3         4
## 1 0.2 0.0 0.0000000 0.0000000
## 2 0.0 0.2 0.0000000 0.0000000
## 3 0.0 0.0 0.1666667 0.0000000
## 4 0.0 0.0 0.0000000 0.1666667
```

A gives us representation of the effective number of replicates for each treatment; the total number of replicates minus an adjustment for observations made in common with other treatments (or, alternately, treatment observations confounded by block effects.)

```
A <- t_delta + N %*% v_delta_inv %*% t(N)
Ainv <- pseudoinverse(A)
```

For means comparisons, we divide our error term, σ by the effective replication

```
ems <- 1.314641
varCov <- Ainv * ems
varCov
```

```
##             [,1]        [,2]        [,3]        [,4]        [,5]
## [1,]  0.58147583 -0.02528156 -0.02528156 -0.02528156 -0.02528156
## [2,] -0.02528156  0.29832238 -0.03033787 -0.03033787 -0.03033787
## [3,] -0.02528156 -0.03033787  0.29832238 -0.03033787 -0.03033787
## [4,] -0.02528156 -0.03033787 -0.03033787  0.29832238 -0.03033787
## [5,] -0.02528156 -0.03033787 -0.03033787 -0.03033787  0.29832238
## [6,] -0.02528156 -0.03033787 -0.03033787 -0.03033787 -0.03033787
##             [,6]
## [1,] -0.02528156
## [2,] -0.03033787
## [3,] -0.03033787
## [4,] -0.03033787
## [5,] -0.03033787
## [6,]  0.29832238
```

```
vcov(lmer(obs ~ trt + (1 | rep), data=Table5.7))
```

```
## 6 x 6 Matrix of class "dpoMatrix"
##             (Intercept)       trt2       trt3       trt4       trt5
## (Intercept)   0.9328203 -0.7101096 -0.7101096 -0.7101096 -0.7101096
## trt2         -0.7101096  1.0397308  0.7101096  0.7101096  0.7101096
## trt3         -0.7101096  0.7101096  1.0397308  0.7101096  0.7101096
## trt4         -0.7101096  0.7101096  0.7101096  1.0397308  0.7101096
## trt5         -0.7101096  0.7101096  0.7101096  0.7101096  1.0397308
## trt6         -0.7101096  0.7101096  0.7101096  0.7101096  0.7101096
##                   trt6
## (Intercept) -0.7101096
## trt2         0.7101096
## trt3         0.7101096
## trt4         0.7101096
## trt5         0.7101096
## trt6         1.0397308
```

This formulation introduces an approximation for the effective number of replicates (see (Steel and Torrie 1960), p 213) when data are unbalanced and increases the error term when means with missing observations are compared. A similar method is used for Tukey's HSD.

For multiple range tests, the harmonic mean of treatment replicates (Milliken and Johnson 1992) is used as an "effective number of replicates" term. For example, the critical value for Duncan's MRT is $HSD_{i,j} = q(\alpha_p, t, \nu) \times \sigma\sqrt{\tilde{n}}$ where $\tilde{n} = t(1n_1 + 1n_2 + 1n_t)^{(} - 1)$ The individual n_t values are taken from the main diagonal of $X_1^t X_1$.

For comparison, we can use the 'agricolae' package to provide LSD and HSD tests.

```
library(agricolae)
LSD.test(aov(tbl5.7.lm),trt="trt")
HSD.test(aov(tbl5.7.lm),trt="trt")
```

We can also use `multcomp` for all pairwise comparison, comparable to Tukey's test,

```
comp <- glht(aov(tbl5.7.lm), linfct = mcp(trt = "Tukey"))
summary(comp)
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: aov(formula = tbl5.7.lm)
##
## Linear Hypotheses:
##            Estimate Std. Error t value Pr(>|t|)
## 2 - 1 == 0   0.9895     1.0255   0.965   0.9201
## 3 - 1 == 0   1.3970     1.0255   1.362   0.7449
## 4 - 1 == 0   0.9570     1.0255   0.933   0.9297
## 5 - 1 == 0   1.6520     1.0255   1.611   0.6030
## 6 - 1 == 0   3.1345     1.0255   3.056   0.0772 .
## 3 - 2 == 0   0.4075     0.8107   0.503   0.9951
## 4 - 2 == 0  -0.0325     0.8107  -0.040   1.0000
## 5 - 2 == 0   0.6625     0.8107   0.817   0.9585
## 6 - 2 == 0   2.1450     0.8107   2.646   0.1523
## 4 - 3 == 0  -0.4400     0.8107  -0.543   0.9930
## 5 - 3 == 0   0.2550     0.8107   0.315   0.9995
## 6 - 3 == 0   1.7375     0.8107   2.143   0.3219
## 5 - 4 == 0   0.6950     0.8107   0.857   0.9496
## 6 - 4 == 0   2.1775     0.8107   2.686   0.1427
## 6 - 5 == 0   1.4825     0.8107   1.829   0.4787
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

```
cld(comp)
```

```
##   1   2   3   4   5   6
## "a" "a" "a" "a" "a" "a"
```

How does ARM compute Treatment (adjusted) Sums of Squares?

Once we've obtained we can partition variance (as sums of squares) using common matrix algebra formula, adapted from (Schabenberger and Pierce 2001), p.98:

Table 2: Partitioning variance from linear models.

Source	DF	SS
Model	$rank(X) - 1$	$\hat{\beta}X'Y - n\bar{Y}^2$
Error	$n - rank(X)$	$Y'Y - \hat{\beta}X'Y$
Total	n	$Y'Y - n\bar{Y}^2$

where $nY\bar{Y}^2$ is a correction factor for the mean and rank(X) is the number of columns (m) in X. When $\hat{\beta}$ is estimated using SVD, we use the effective rank of X, which does not include aliased (redundant) columns. This simplifies working with missing data. To partition sums of squares, we find the model SS for a reduced model then subtract. For example, to compute replicate sums of squares, we fit to both $RCB(X = X_0|X_1|X_2)$ and $CRD(X = X_0|X_1)$ models. Since the RCB model includes both treatment and replicate effects, while the CRD model contains treatments only, the difference in model SS of the two is attributable to replicates,

$$\begin{aligned} Model_{CRD}SS &= SS_{Trt} \\ Model_{RCB}SS &= SS_{(Trt+Rep)} \\ SS_{Rep} &= Model_{RCB}SS - Model_{CRD}SS \end{aligned}$$

However, if the data are not balanced, then there will be some confounding among effects. If we fit a block-only model ($X = X_0|X_2$), the model SS will not equal as found by subtraction above. Remember that the effect of specific block is calculated based on observations taken from treatments appearing in that block; a block effect will in inflated if a low-scoring treatment is missing (or vice-versa). Thus, to provide a treatment SS, adjusted for confounding replicate effects, we must also estimate a block-only model,

$$SS_{Trt(adj)} = Model_{RCB}SS - Model_{Block}SS$$

How are non-RCB designs modeled in ARM?

CRD designs lack a blocking structure, so no X_2 matrix is required. Thus, treatment least square means are identical to arithmetic means, and it is not possible to compute an adjusted treatment SS. Latin squares include an additional blocking matrix (X_3) associated with columns; least square means and adjusted treatment SS are computed using this additional matrix. Lattice designs also include additional blocking matrices for incomplete blocks in replicates, and for lattice squares, incomplete columns within replicates.

However, since by design these matrices represent random effects, adjusted treatment means include an additional analysis step, the recovery of intrablock information as described in (John and Williams 1995).

References

Giesbrecht, Francis G, and Marcia L Gumpertz. 2004. *Planning, Construction, and Statistical Analysis of Comparative Experiments.* Wiley-Interscience.

John, J.A., and E.R. Williams. 1995. *Cyclic and Computer Generated Designs.* Chapman; Hall/CRC.

Milliken, George A., and Dallas E. Johnson. 1992. *Analysis of Messy Data.* 1st ed. Vol. Designed Experiments. Chapman; Hall/CRC.

Schabenberger, Oliver, and Francis J. Pierce. 2001. *Contemporary Statistical Models for the Plant and Soil Sciences [Hardcover].* CRC Press.

Steel, Robert G D, and James H Torrie. 1960. *Principles and Procedures of Statistics, a Biometrical Approach.* Second. McGraw-Hill.