

Latin Square Designs

Bret Larget

Departments of Botany and of Statistics
University of Wisconsin—Madison

March 27, 2007

The Big Picture

- A *Latin Square* design is an example of an *incomplete block design* where there is a single treatment and two blocking variables, each with the same number of levels.
- Only a single treatment is applied within each combination of blocking variables.
- For example, when there are four treatments labeled A, B, C, and D and blocking variables we will call the “row effect” and the “column effect”, an example Latin Square looks like this.

A	B	C	D
B	C	D	A
C	D	A	B
D	A	B	C

- Each letter appears in each row and in each column exactly once.
- Latin squares designs are often used when each subject (row) is measured for each treatment level where time (or period) is another blocking variable.

Why use a Latin Square?

- Latin square designs are reasonable choices when it is impossible to use each treatment level for the same combination of blocking levels.
- For example, consider an experiment with four diets, each to be given to four cows in succession.
- If each cow was given the diets in the same order, the treatment effect would be *confounded* with the effect due to the order in which the diets were given.
- Each cow can only be given a single diet during a single time period.
- It is common to use multiple Latin squares in a single experiment (for example if you had 8 or 12 or 16 cows to allocate).

Choosing a Latin Square at Random

- There are many different possible Latin Squares of a given size.
- There is not a simple way to count all possible Latin squares of a given size or to pick one perfectly at random.
- For any Latin square, swapping the order of the rows or the columns provides a different Latin square.
- The R function shown next begins with a standard Latin square and then performs many random permutations of rows and columns to pick one at random.

R Code

```

> source("latin.R")
> latin

function (n, nrand = 20)
{
  x = matrix(LETTERS[1:n], n, n)
  x = t(x)
  for (i in 2:n) x[i, ] = x[i, c(i:n, 1:(i - 1))]
  if (nrand > 0) {
    for (i in 1:nrand) {
      x = x[sample(n), ]
      x = x[, sample(n)]
    }
  }
  x
}

```

Examples

```

> latin(4)
      [,1] [,2] [,3] [,4]
[1,] "C"  "D"  "A"  "B"
[2,] "B"  "C"  "D"  "A"
[3,] "A"  "B"  "C"  "D"
[4,] "D"  "A"  "B"  "C"

> latin(4)
      [,1] [,2] [,3] [,4]
[1,] "B"  "A"  "C"  "D"
[2,] "A"  "D"  "B"  "C"
[3,] "D"  "C"  "A"  "B"
[4,] "C"  "B"  "D"  "A"

> latin(8)
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
[1,] "H"  "E"  "G"  "D"  "A"  "F"  "B"  "C"
[2,] "A"  "F"  "H"  "E"  "B"  "G"  "C"  "D"
[3,] "G"  "D"  "F"  "C"  "H"  "E"  "A"  "B"
[4,] "C"  "H"  "B"  "G"  "D"  "A"  "E"  "F"
[5,] "D"  "A"  "C"  "H"  "E"  "B"  "F"  "G"
[6,] "E"  "B"  "D"  "A"  "F"  "C"  "G"  "H"
[7,] "B"  "G"  "A"  "F"  "C"  "H"  "D"  "E"
[8,] "F"  "C"  "E"  "B"  "G"  "D"  "H"  "A"

```

Model



$$y_{ijk} = \mu + \alpha_i + \beta_j + \gamma_k + e_{ijk}$$

where

- μ is a population mean.
- α_i is the main effect of treatment i , $i = 1, \dots, n$,
 $\sum_{i=1}^n \alpha_i = 0$.
- β_j is a main effect of the first blocking variable j , $j = 1, \dots, n$,
 $\sum_{j=1}^n \beta_j = 0$.
- γ_k is a main effect of the second blocking variable k , $k = 1, \dots, n$,
 $\sum_{k=1}^n \gamma_k = 0$.
- $e_{ijk} \sim \text{iid } N(0, \sigma_e^2)$ is the subplot (individual) error distribution.
- Each pair of effects is crossed, but there is not a three-way cross.
- Often, the row corresponds to a subject and is modeled as a random effect.

Example

```
> millet = read.table("millet.txt", header = T)
> str(millet)

'data.frame': 25 obs. of 4 variables:
 $ Row      : Factor w/ 5 levels "R1","R2","R3",...: 1 1 1 1 1 2 2 2 2 2 ...
 $ Column   : Factor w/ 5 levels "C1","C2","C3",...: 1 2 3 4 5 1 2 3 4 5 ...
 $ Treatment: Factor w/ 5 levels "A","B","C","D",...: 2 5 1 3 4 4 1 5 2 3 ...
 $ Yield    : int  253 226 285 283 188 255 293 265 290 260 ...

> attach(millet)
> millet.lm = lm(Yield ~ Row + Column + Treatment)
> anova(millet.lm)
```

Analysis of Variance Table

```
Response: Yield
      Df Sum Sq Mean Sq F value Pr(>F)
Row      4 14256.6   3564.1   3.3764 0.04531 *
Column   4  6906.2   1726.5   1.6356 0.22900
Treatment 4  4156.6   1039.1   0.9844 0.45229
Residuals 12 12667.3   1055.6
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Summary

```
> summary(millet.lm)

Call:
lm(formula = Yield ~ Row + Column + Treatment)

Residuals:
    Min     1Q   Median     3Q      Max
-44.68 -12.48   1.12  16.52  54.92

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  236.08     23.43  10.076  3.3e-07 ***
RowR2         25.60     20.55   1.246  0.2366
RowR3         3.00     20.55   0.146  0.8863
RowR4        -29.80     20.55  -1.450  0.1726
RowR5         40.00     20.55   1.947  0.0754 .
ColumnC2      24.40     20.55   1.187  0.2580
ColumnC3      45.80     20.55   2.229  0.0457 *
ColumnC4      44.40     20.55   2.161  0.0516 .
ColumnC5      28.20     20.55   1.372  0.1951
TreatmentB    -7.00     20.55  -0.341  0.7393
TreatmentC   -17.40     20.55  -0.847  0.4137
TreatmentD   -31.60     20.55  -1.538  0.1500
TreatmentE   -32.20     20.55  -1.567  0.1431
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 32.49 on 12 degrees of freedom
Multiple R-Squared:  0.6665, Adjusted R-squared:  0.3331
F-statistic: 1.999 on 12 and 12 DF,  p-value: 0.1223
```

Plots

Use R live.