Latin squares and related experimental designs

1 Introduction

A Latin square is an arrangement of v copies of v symbols into a $v \times v$ square so that (i) each symbol occurs once in each row, and (ii) each symbol occurs once in each column. Three distinct Latin squares of order $v = 4$ are shown in Example 1. Other than for small v, the number of distinct (non-identical as matrices) Latin squares is not generally known, though it is known that it grows rapidly with v. For $v = 3$ the number of distinct Latin squares is 12, for $v = 7$ is greater than 6.1×10^{13}, and for $v = 11$ is greater than 7.7×10^{47}; see Laywine and Mullen (1998, chapter 1).

Our interest in Latin squares is due to their many uses in designing experiments. Suppose you have the task of comparing wear of drill bits of four different compositions when used in industrial presses. Each day each of your four press operators will be given a new bit, with wear to be measured at day's end; this will be repeated for four days. A Latin square can be employed to block out variation associated with operators and with days. Consider the first square in Example 1, and identify rows with operators, columns with days, and symbols with bit compositions. Then operator #1 uses composition #1 on day #1, operator #2 uses composition #3 on day #4, and so on. This design displays a basic fairness of allocation, in that each composition is exposed once to each operator, and once to each day. Statistically this implies orthogonal estimation of the operator, day, and composition effects. This is but one example of the many adaptations of Latin squares for obtaining good experimental designs that will be explored in this chapter.

Before proceeding, one more basic definition is needed. Two Latin squares of order v are said to be *orthogonal* (sometimes called a *Graeco-Latin* pair) if, upon superimposing one square upon the other, every ordered pair of symbols occurs in the cells exactly once. Check Example 1 and you will see that every pair of squares has this property; the superimposition of all three squares is displayed in Example 2.I. The set of three squares is *mutually orthogonal.* This combinatorial orthogonality of Latin squares translates into statistical orthogonality of various treatment and blocking factors when exploited in an experimental design. Latin square designs are valuable because they provide orthogonal estimation, and because they do so with with relatively little experimental material for given numbers of factors and levels.

2 Latin squares for blocking out two sources of nuisance variation

The example in section 1 illustrates the application of a Latin square as a row-column design for eliminating two sources of nuisance variation. Given a Latin square, let rows denote levels of one blocking factor, columns the levels of a second blocking factor, and symbols the levels of your treatment factor. The result is simultaneously a complete block design (see chapter ???) for treatments with respect to the row blocking factor and with respect to the column blocking factor.

Let $d[i, j]$ be the treatment appearing in row i, column j of Latin square d (e.g. $d[2, 4] = 3$ for the first square of Example 1). Denote by Y_{ij} the observation for the run in row i, column j (cell i, j). Analysis of the data is based on this model:

$$Y_{ij} = \mu + \rho_i + \gamma_j + \tau_{d[i,j]} + E_{ij} \tag{1}$$

inducing the skeleton ANOVA (put $s = 1$) in Table 1.I. For ANOVA details, see chapter ???

Use of a Latin square means that each pair of rows, columns, and treatments is statistically orthogonal, one consequence of which is that estimation of treatment effects is based on treatment data means only. A Latin square thus provides a simple analysis while removing variation due to two v-level nuisance factors from comparisons of a v-level treatment factor. It does this using only v^2 experimental units, far less than the v^3 possible combinations of rows, columns, and treatments. This is a highly efficient use of resources.

The drawbacks to this use of a Latin square arise from the requirement that numbers of rows, columns, and treatments are all equal. What can one do, maintaining orthogonality, if this restriction cannot be met, or if it leaves too few degrees of freedom for error? These questions are addressed in the following subsections.

2.1 More rows and columns

More levels of the row factor and/or column factor can be accommodated by juxtaposing several Latin squares. Given ab Latin squares of order v, arrange them into a rows of b squares each, then "paste" them together to yield an $av \times bv$ row-column design. Any ab Latin squares can be used (even ab copies of the same square). This technique (i) allows some flexibility in numbers of rows and columns while (ii) preserving the statistical orthogonality properties of a single Latin square and (iii) providing more degrees of freedom for error. If now d denotes the $av \times bv$ design, then the model is still (1), and the skeleton ANOVA becomes that in Table 1.II. It is also possible to extract degrees of freedom for treatment$\times$row and/or treatment$\times$column interaction, but how much can be done is design specific, that is, depends on the Latin squares that are pasted together.

2.2 More squares

Another possibility for gaining error degrees of freedom is multiple Latin squares. If the experiment described in section 1 is to be run at m sites, each with its own operator and day-to-day variability, then m Latin squares (identical or different) can be employed. The resulting skeleton ANOVA is in Table 1.VI (put $s = 1$). Again, the orthogonality properties of a single Latin square are preserved, so that treatment estimation is based on means.

2.3 More EUs per cell; semi-Latin squares

The basic Latin square has one experimental unit (EU, run), and thus only one treatment evaluated, per cell. If we can procure $k > 1$ EUs per cell in an $n \times n$ row-column blocking layout, then $v = nk$ treatments can be accommodated in each row and column. The design is a *semi-Latin square*: there are k symbols (treatments) in each cell so that each symbol appears once in each row and once in each column.

Semi-Latin squares can be found as superimpositions of k Latin squares of order n, each square having its own set of distinct symbols. Choice of the k squares depends on which of two reasonable models are employed. For Y_{ijl} the measurement on unit l in cell (i, j), they are

$$Y_{ijl} = \mu + \rho_i + \gamma_j + \tau_{d[i,j,l]} + E_{ijl} \tag{2}$$

and

$$Y_{ijl} = \mu + \beta_{ij} + \tau_{d[i,j,l]} + E_{ijl} \tag{3}$$

Model (2) is the direct extension of (1), accounting for noise through additive effects for the row and column nuisance factors. The more detailed model (3) fits a separate noise effect for each cell; this is tantamount to allowing interaction of the row and column factors. The respective ANOVA skeletons are Table 1.III and IV. With model (2) treatment analysis is based on means, a consequence of the orthogonality of treatments to rows and columns. Because treatments need not be orthogonal to cells, treatment analysis with model (3) is adjusted for the cell blocks. There are also fewer error degrees of freedom with model (3). These are the tradeoffs for the further variance reduction afforded by the finer model (3).

With model (2), any k Latin squares may be superimposed to form the semi-Latin square, for all such choices provide orthogonality of rows, columns, and symbols and so are equally efficient. With model (3) the driving consideration is not the rows or the columns, but the cells. The goal is to maximize the efficiency of the block design formed by the cell blocks. So long as $k < n$, the best choice is to superimpose k MOLS. A semi-Latin square formed in the way is called a *Trojan square*. Example 2.II is a Trojan square (corresponding to Example 2.I with symbols distinguished). Details on maximal k for $n \times n$ Trojan squares are in section 3. For $n = 6$, the best semi-Latin square for $k = 2$ is reported in Bailey and Royle (1997). Bailey (1992) gives a comprehensive treatment of semi-Latin squares.

Latin and semi-Latin squares use each treatment once in each row and each column. With k EUs per cell, orthogonality in an $n \times n$ row-column design is maintained if each treatment appears nk/v times in each row and column. A *k-depth Latin square* is an arrangement of vk copies of v symbols into a $v \times v$ square so that each cell has k symbols, and each symbol occurs k times in each row and in each column.

If when superimposing k Latin squares in the semi-Latin construction above, one takes the symbols to be the same in each square, then one gets a k-depth Latin square (see Example 2.I). This particular class of designs, while fully efficient for model (2), is inefficient for (3) due to replication of the same treatment within cell blocks. The skeleton ANOVA for model (2) is Table 1.I with $s = 1$ and an additional $(k-1)v^2$ degrees of freedom added to error and total.

For model (3), start with a $n \times n$ Latin square and a BIBD (chapter ???) for v treatments in n blocks of size k. Place block i in each cell of the Latin square containing symbol i for $i = 1, \ldots, n$. The resulting $n \times n$ row-column design with k runs per cell is fully efficient for both models (2) and (3). It is a k-depth Latin square if $n = v$. Example 2.III displays $n = v = 3$, $k = 2$. The standard BIBD analysis applies.

2.4 Other numbers of rows and columns: Youden designs

If one drops a single row from a Latin square, the rows of the resulting $(v-1) \times v$ array are a complete block design, while the columns are a BIBD. This *incomplete Latin square* is one version of a class of designs known as *Youden squares*. A Youden square is a row-column design with k rows and v columns for which each treatment appears once in each row, and columns are blocks of a BIBD. Every Youden square is an incomplete Latin square, though the converse is not true.

Rows and columns may be added by joining separate Youden designs and Latin squares. Paste columns of t Youden squares into a $k \times tv$ array, then add a pasting of at Latin squares as in section 2.1, to produce a $(k + av) \times tv$ row-column design called a *regular generalized Youden design*. With the standard row-column model (1) the skeleton ANOVA is Table 1.V. Since treatments are orthogonal to rows, the treatment adjustment is for columns only.

Here is the general idea. Start with any incomplete block design, with v treatments in b blocks of size k. Thinking of the blocks as columns, paste them into a $k \times b$ row-column array. If the treatments are equally replicated, and if b is a multiple of v (so $b = tv$ for some t), then the treatments can be ordered within the columns so that each treatment appears t times in each row. This is now a row-column design with treatments orthogonal to rows; treatment estimates are the same as for the original block design without the row factor. If more rows are needed, it can be extended to a $(k + av) \times tv$ row-column design using Latin squares as explained above and with the same skeleton ANOVA.

3 Designs based on MOLS

The Trojan squares of section 2.3 are built using mutually orthogonal Latin squares, or MOLS. Here other designs based on or related to MOLS will be described. The common theme is that of adding more factors to the designs of section 2, maintaining orthogonality without changing the amount of experimental material. But first a basic question must be addressed: how many MOLS of a given order are there? Let $N(v)$ be the largest number of MOLS of order v. Then $N(v) \leq v - 1$ for every v, and $N(v) = v - 1$ for v that is a prime number or a power of a prime number. Example 1 provides $N(4) = 3$ MOLS of order 4. While upper bounds sharper than $v-1$ are known for some other v, and $N(v) \geq 2$ for all $v > 6$, determining the exact value of $N(v)$ is a daunting (to say the least) combinatorial problem. For $v \leq 20$ not a prime power, it is known that $N(6) = 1$, $N(10) \geq 2$, $N(12) \geq 5$, $N(14) \geq 3$, $N(15) \geq 4$, $N(18) \geq 3$, and $N(20) \geq 4$. For a catalog of known MOLS for every $v \leq 56$ along with further results, see Abel, Colbourn, and Dinitz (2006).

3.1 Row-column designs with more blocking and/or treatment factors

Let's introduce a third blocking factor, four suppliers of parts to be drilled, to the drill bit experiment described in section 1. The original design, with rows for operators, columns for days, and symbols for drill bits, is the first Latin square in Example 1. Let the levels of supplier be the symbols in the second square. Superimposing the two squares, we see, for instance, that operator 1 on day 2 uses drill bit composition 3 on parts from supplier 2. The pair of MOLS means that each supplier's parts are used once with each bit type, once on each day, and once by each operator. That is, the supplier factor is orthogonal to the three factors operator, day, and bit type (which, as we already know, are orthogonal to one another). The third blocking factor has been introduced without requiring any further experimental runs, and without changing that treatment effects are estimated by data means.

The example shows that two superimposed MOLS provide a design for comparing v treatments using v^2 runs subject to three v-level blocking factors. In the same way, s superimposed MOLS provide a design comparing v treatments using v^2 runs subject to $s+1$ v-level blocking factors; effects of all factors are orthogonal to one another. For the drill bit example, the symbols in the third square of Example 1, used as levels of a fourth blocking factor, produce the design in Example 2.I.

Another advantage of the orthogonality is that any of the blocking factors could alternatively be used as a treatment factor, so long as estimation of treatment interactions is not required. Thus s MOLS are designs for f v-level treatment factors using v^2 runs subject to $s+2-f$ v-level blocking factors, for any $1 \leq f \leq s+2$. Table 1.I shows the skeleton ANOVA. Each of the first $s+2$ rows is identified with either a blocking factor or a treatment factor. Because only main effects, and no interactions, of treatment factors are estimable, these are called (blocked) *main effects plans*. The case $f = s+2$ demonstrates that s MOLS are equivalent to a strength two orthogonal array (see chapter ???) with v^2 runs on $s+2$ v-level factors.

3.2 More rows and columns

The designs of section 2.1 can also incorporate more blocking (or treatment) factors through orthogonal mates. An $av \times bv$ array with one symbol per cell is an *F-rectangle* (*F-square* if $a = b$) if each of v symbols occurs b times in each row and a times in each column; the designs of section 2.1 are F-rectangles. Two F-rectangles are orthogonal if superimposition produces each ordered pair of symbols in ab cells. Mutually orthogonal F-rectangles admit the same applications as described for MOLS in section 3.1, with now the row and column factors having av and bv levels (and thus $av-1$ and $bv-1$ df) respectively. A set of s mutually orthogonal F-rectangles is equivalent to a mixed level, strength 2 orthogonal array with $s+2$ columns. For examples of mutually orthogonal F-squares, and for further references, see Laywine and Mullen (2006).

3.3 More squares

Orthogonal mates can likewise be found for sets of m Latin squares in section 2.2. A collection of s ordered sets, each containing m Latin squares of order v, is an (s, m) *orthogonal collection* if, on superimposition of the m corresponding squares of any two sets, each ordered pair of symbols occurs m times. MOLS are the special case $m = 1$.

The symbols in a given set of m squares, the same for each of these m squares, are levels of additional v-level treatment factors. With $s = 2$ and $m = 3$, for example, the experiment of section 1 could be run at 3 sites, each with its own days and operators, but evaluating levels of two treatment factors: drill bit composition and (say) lubricant. The skeleton ANOVA is Table 1.VI.

Let $N_m(v)$ be the largest s for an (s, m) collection of order v. Then $N(v) \leq N_m(v) \leq m(v-1)$. Known values of $N_m(v)$ for small m and v are $N_2(2) = 2$, $N_3(2) = 1$, $N_4(2) = 4$, $N_2(3) = 2$, $N_3(3) = 6$, $N_4(3) = 2$, $N_2(4) = 5$, $N_3(4) = 4$, $N_4(4) = 12$, $N_2(5) \geq 4$, $N_3(5) \geq 6$, $N_4(5) \geq 5$. Also, $N_2(6) \geq 4$, that is, at least four orthogonal six-level factors can be accommodated in two 6×6 squares, while $N_1(6) \equiv N(6) = 1$. Designs and further details may be found in Morgan (1998).

3.4 More EUs per cell

It is not possible to add orthogonal v-level treatment factors to semi-Latin squares, for they have less than v^2 experimental units. However, this can be done for the other designs in section 2.3.

Starting with an (s, k)-orthogonal collection of Latin squares of order v, we can make a $v \times v$ row-column design with k units per cell for s orthogonal treatment factors. The symbol in cell (i, j) of square u $(= 1, \ldots, k)$ in set w $(= 1, \ldots, s)$ of the collection is the level of factor w that appears on unit u in cell (i, j) of our design. If $s = 1$ this is a k-depth Latin square of section 2.3; if $k = 1$ it is the design of section 3.1. Relative to the k-depth square, $s - 1$ additional orthogonal treatment factors have been incorporated; relative to a set of MOLS, incorporating k EUs per cell has increased error degrees of freedom by $(k-1)v^2$. The skeleton ANOVA is Table 1.I with an additional $(k-1)v^2$ degrees of freedom added to error and total. Example 2.IV has five 4-level factors in a 4×4 design with $k = 2$ units per cell. Like a single k-depth Latin square, this design is fully efficient for model (2) but inefficient for model (3).

The design in the last paragraph of section 2.3 can accommodate orthogonal treatment factors by replacing the BIBD with an orthogonal set of BIBDs. Orthogonal sets of BIBDs may be found in Morgan and Uddin (1996). The orthogonality of treatment factors is after adjusting for the cell blocking factor in model (3).

3.5 Orthogonal Youden designs

It is sometimes possible to add orthogonal treatment factors to regular generalized Youden designs. Here, too, the orthogonality of treatment factors is after adjusting for the column blocks. See Morgan and Uddin (1996).

4 Discussion

This paper provides an overview of the chief uses of Latin squares for designs for industrial and manufacturing experiments. The designs combine orthogonal estimation with an economy of experimental units, providing for both blocking and treatment factors. While the variants presented offer flexibility in numbers of rows, columns, factors, etcetera, there are many experimental possibilities that are not amenable to the inherent restrictions of Latin squares. In such cases orthogonality is sacrificed.

Specialized Latin squares have been employed in experimental situations not discussed above, including the following. *Row-complete Latin squares*, including *Williams squares*, are used as crossover designs (also called changeover or repeated measures designs); see Hinkelmann and Kempthorne (2005, chapter 19). *Quasi-complete Latin squares* are row-column designs with non-directional neighbor balance as discussed in Bailey (1984) and Morgan (1988). Hedayat (1973) examines designs based on *self-orthogonal Latin squares*. *Gerechte squares* (Bailey, Kunert, and Martin, 1990) are Latin squares with a third blocking factor orthogonal to treatments but not to rows or columns. When $v = 9$ they are popularly known as completed *Sudoku squares*.

REFERENCES

ABEL, R. J. R., COLBOURN, C. J., AND DINITZ, J. H. (2006). Mutually orthogonal Latin squares (MOLS). In *The CRC Handbook of Combinatorial Designs, 2nd Ed.*, Eds. C. J. Colbourn and J. H. Dinitz, CRC Press, Boca Raton.

BAILEY, R. A. (1984). Quasicomplete Latin squares: construction and randomization. *J. Roy. Statist. Soc. Ser. B* **46**, 323–334.

BAILEY, R. A. (1992). Efficient semi-Latin squares. *Statistica Sinica* **2**, 413–437.

BAILEY, R. A., KUNERT, J., AND MARTIN, R. J. (1990). Some comments on Gerechte designs I: Analysis for uncorrelated errors. *J. Agron. Crop Sci.* **165**, 121–130.

BAILEY, R. A. AND ROYLE, G. (1997). Optimal semi-Latin squares with side six and block size two. *Proc. Roy. Soc. London Ser. A* **453**, 1903–1914.

HEDAYAT, A. (1973). Self orthogonal Latin square designs and their importance. *Biometrics* **29**, 393–396.

HINKELMANN, K. AND KEMPTHORNE, O. (2005). *Design and analysis of experiments, Vol. 2: Advanced experimental design*. John Wiley & Sons, New York.

LAYWINE, C. F. AND MULLEN, G. L. (1998). *Discrete mathematics using Latin squares*. John Wiley & Sons, New York.

LAYWINE, C. F. AND MULLEN, G. L. (2006). Frequency squares and hypercubes. In *The CRC Handbook of Combinatorial Designs, 2nd Ed.*, Eds. C. J. Colbourn and J. H. Dinitz, CRC Press, Boca Raton.

MORGAN, J.P. (1988). Balanced polycross designs. *J. Roy. Statist. Soc. Ser. B* **50**, 93–104.

MORGAN, J. P. (1998). Orthogonal collections of Latin squares. *Technometrics* **40**, 327–333.

MORGAN, J.P. AND UDDIN, N. (1996). Optimal blocked main effects plans with nested rows and columns and related designs. *Annals of Statistics* **24**, 1185–1208.

Example 1. *Three 4×4 Latin squares. These squares are mutually orthogonal.*

1	2	3	4
2	1	4	3
3	4	1	2
4	3	2	1

1	2	3	4
3	4	1	2
4	3	2	1
2	1	4	3

1	2	3	4
4	3	2	1
2	1	4	3
3	4	1	2

Example 2. *Various designs based on Latin squares.*

I :

$1,1,1$	$2,2,2$	$3,3,3$	$4,4,4$
$2,3,4$	$1,4,3$	$4,1,2$	$3,2,1$
$3,4,2$	$4,3,1$	$1,2,4$	$2,1,3$
$4,2,3$	$3,1,4$	$2,4,1$	$1,3,2$

II :

$1,5,9$	$2,6,10$	$3,7,11$	$4,8,12$
$2,7,12$	$1,8,11$	$4,5,10$	$3,6,9$
$3,8,10$	$4,7,9$	$1,6,12$	$2,5,11$
$4,6,11$	$3,5,12$	$2,8,9$	$1,7,10$

III :

$1,2$	$2,3$	$1,3$
$1,3$	$1,2$	$2,3$
$2,3$	$1,3$	$1,2$

IV :

$(1,4,2,3,3),(3,3,3,3,3)$	$(2,2,3,4,1),(4,1,2,4,1)$	$(3,1,1,2,4),(1,2,4,2,4)$	$(4,3,4,1,2),(2,4,1,1,2)$
$(2,3,1,4,4),(4,4,4,4,4)$	$(1,1,4,3,2),(3,2,1,3,2)$	$(4,2,2,1,3),(2,1,3,1,3)$	$(3,4,3,2,1),(1,3,2,2,1)$
$(3,2,4,1,1),(1,1,1,1,1)$	$(4,4,1,2,3),(2,3,4,2,3)$	$(1,3,3,4,2),(3,4,2,4,2)$	$(2,1,2,3,4),(4,2,3,3,4)$
$(4,1,3,2,2),(2,2,2,2,2)$	$(3,3,2,1,4),(1,4,3,1,4)$	$(2,4,4,3,1),(4,3,1,3,1)$	$(1,2,1,4,3),(3,1,4,4,3)$

(I)

Source	df
rows	$v-1$
columns	$v-1$
symbols in square 1	$v-1$
$\vdots$	$\vdots$
symbols in square s	$v-1$
error	$(v-1)(v-s-1)$
total	v^2-1

(II)

Source	df
rows	$av-1$
columns	$bv-1$
treatments	$v-1$
error	$(av-1)(bv-1)-(v-1)$
total	abv^2-1

(III)

Source	df
rows	$n-1$
columns	$n-1$
treatments	$nk-1$
error	$(nk-2)(n-1)$
total	n^2k-1

(IV)

Source	df
cells	n^2-1
treatments (adj)	$nk-1$
error	$(nk-n-1)(n-1)$
total	n^2k-1

(V)

Source	df
rows	$k+av-1$
columns	$tv-1$
treatments (adj)	$v-1$
error	$(tv-1)(k+av-1)-(v-1)$
total	$tv(k+av)-1$

(VI)

Source	df
squares	$m-1$
rows(squares)	$m(v-1)$
columns(squares)	$m(v-1)$
symbols in set 1	$v-1$
$\vdots$	$\vdots$
symbols in set s	$v-1$
error	$(v-1)(v-s-1)+(m-1)(v-1)^2$
total	mv^2-1

Table 1: ANOVA skeletons for various applications of Latin squares (see text)