TOOLS FOR THE CONSTRUCTION OF EFFECTIVE EXPERIMENTAL DESIGNS

M. F. Franklin

R. W. Payne

Follow this and additional works at: http://newprairiepress.org/agstatconference

Part of the Agriculture Commons, and the Applied Statistics Commons

This work is licensed under a Creative Commons Attribution-Noncommercial-No Derivative Works 4.0 License.

Recommended Citation


This is brought to you for free and open access by the Conferences at New Prairie Press. It has been accepted for inclusion in Conference on Applied Statistics in Agriculture by an authorized administrator of New Prairie Press. For more information, please contact cads@k-state.edu.
TOOLS FOR THE CONSTRUCTION
OF EFFECTIVE EXPERIMENTAL DESIGNS

M. F. Franklin and R. W. Payne

Scottish Agricultural Statistics Service, Bucksburn, Aberdeen, AB2 9SB, Scotland
AFRC Institute of Arable Crops Research, Rothamsted, Harpenden, AL5 2JQ, England

Abstract

Experimenters should be able to choose between competing designs the one which yields the required information clearly and efficiently at the desired precision. One way to achieve this is to allow interaction between design and analysis but few statistical analysis packages include more than rudimentary design facilities. We review some of the theory and tools for design construction with a view to providing the statistician and experimenter with a tool-kit for building the most effective design. Examples in the design process are techniques for determining aliases and patterns of confounding, algorithms for constructing fractional factorial and incomplete block designs and methods of (restricted) randomization. Examples in analysis include algorithms for calculating efficiency factors, for estimating variance components and for assessing general balance.
1. Introduction

The use of computer packages for statistical analysis is helping slowly to improve the application of statistical methods by agricultural and biological scientists. The use of such packages for the design of experiments will further aid this understanding, particularly if the experimenter can see the effects of changes to the design upon the analysis. Computer construction of designs has many practical advantages including use of superior designs, improved methods of randomization, formulation and testing of the data analysis, avoidance of clerical errors, production of recording forms and patterns for data collection and storage. Until recently few major analysis packages have included routines specifically for constructing designs. Now programs, e.g. Echip, are being marketed which assist in the construction of designs, particularly in industrial applications. These programs usually include special analysis procedures associated with the designs produced. Features of many of these programs have been summarised in a report by Rasch and Darius (1993).

There are marked differences between the types of design commonly used in industry, in agriculture and biology and in medicine. Design programs have usually targeted designs for industry but here we concentrate on those for agricultural and biological research. For these applications the Scottish Agricultural Statistics Service has produced several programs for experimental designs. One, ALPHA, for the production and analysis of alpha or generalised lattice designs has been widely used in crop variety testing systems. Another, DSIGNX, is used to generate a wider class of designs but does not contain an analysis routine. However, the designs created by this program have usually been analysed using Genstat and this link has highlighted the advantage of including the design procedures in an analysis package. We consider some design procedures which could be included in a statistical analysis package such as SAS or Genstat to give the user the ability to construct and analyse a wide range of practically useful designs.

We will not here discuss methods for the production of optimal designs for in our experience experimenters in biological sciences rarely have sufficient knowledge to be able to specify exactly both the model and the criteria they require to select the optimal design. However, they usually have a good understanding of their requirements and seek designs which are efficient and robust. Often they may wish to choose a design known to be good but then to modify it to meet particular requirements. We refer to such efficient practically useful designs as 'effective designs'.
We believe there is a strong case for building design procedures into an analysis package rather than adding analysis procedures to a design program. The main reasons are the power and versatility of the analysis packages, the experience with the use of the package already gained by the experimenter and the existence within the analysis packages of routines which are useful in design also.

2. Features of a statistical analysis package

We summarise some of the facilities found in several statistical analysis packages which can also be used in constructing designs, illustrating these with a few examples. We sometimes refer to specific routines offered by Genstat but similar procedures may be offered by other packages.

In a course on the design of experiments and analysis of variance offered by the Scottish Agricultural Statistics Service, scientists analyse data from a series of increasingly more complex designs. The data they use is simulated and depends on the randomization. Hidden from the participants, the units have pre-set effects - typical of those expected in many forms of biological experiment - as do the treatments. The observation in any cell is formed by \( y_{ij} = \mu + u_i + \tau_j \), where \( u_i \) and \( \tau_j \) are the unit and treatment effects. Scientists can see the extent to which the results from their analyses are affected by different randomizations and how effective use of blocks can both reduce the standard errors for treatment differences and provide more consistent conclusions. In another application, covariates may be used to define features of the experimental material which cannot readily be eliminated by blocking. Randomization followed by a 'dummy' analysis of covariance (i.e. one in which an artificial variable is analysed) allows the experimenter to reject randomizations which do not meet pre-set criteria for the covariance efficiency factors. In a third application an experimenter may have available a design to which an extra constraint is to be added e.g. the scientist has available an efficient incomplete block design but would like the design to still be efficient if it is later decided that, say, two-way blocking or adjustment for neighbours is required. Randomization followed by dummy analysis of variance can help the experimenter avoid poor designs.

As one would expect from sophisticated packages, most can be used as programming languages and have all the features such as loops, procedures and structure definition and handling that are normally required by the user. Genstat distinguishes between factors and variates but allows calculations to be based on either and it allows data to be transferred between structures of different types. It can be used in an interactive mode and user-friendly interfaces can be added. It includes three directives Generate, Restrict and Randomize which are directly useful for building designs.
One of the particular strengths of Genstat is the powerful algorithm for analysis of variance (Payne and Wilkinson, 1977; Payne and Tobias, 1992) which uses a series of sweep operations to analyse designs with the property of first-order balance. This allows for a wide range of designs with orthogonal blocking structures which can incorporate very complicated crossing and nesting of factors. In many designs the treatment terms are orthogonal to blocks, but the algorithm can also handle balanced and partially-balanced confounding. Most designs in standard texts such as Cochran and Cox (1957) can be analysed. Genstat automatically determines the strata (or error terms) where each treatment term is estimated, together with the corresponding efficiency factors. The analysis extends to the analysis of covariance, and for each treatment effect covariance efficiency factors are presented. As with other packages various alternative forms of analysis are available for unbalanced designs. Regression-with-factors may be used for designs with a single error stratum and residual maximum likelihood (REML) for designs with multiple error strata. The latter procedure may be used for a variety of mixed-model applications allowing, for example, the recovery of inter-block information. If the data have a non-normal distribution similar analyses may be performed through generalized linear models. Thus for the single error stratum the same regression facilities are available and for multiple error strata procedures have been written to allow the analysis of generalized mixed models. In most packages random numbers from various distributions can be generated so that data simulation is possible. Other relevant important features of packages include powerful routines for handling or modifying structures and for printing or storing designs.

One feature of Genstat requires special mention. In the analysis of variance procedures random and fixed effects are separated. The random effects are considered to be a consequence of the natural structure of the experimental units. This structure is defined by the block structure formula using simple nesting (/) and crossing (*) operations. Treatment structures are defined in the same way. Apart from providing experimenters with a good intuitive understanding of their problems the block formulae determine both the randomization procedure and the error strata in the analysis of variance. For example, a Latin Square with sub-plots has block structure (Row*Column)/Sub-plot giving error strata and randomization procedure:

<table>
<thead>
<tr>
<th>Error strata</th>
<th>Randomization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row</td>
<td>randomize Rows</td>
</tr>
<tr>
<td>Column</td>
<td>randomize Column</td>
</tr>
<tr>
<td>Row.Column</td>
<td>randomize Sub-plots within</td>
</tr>
<tr>
<td>(Row.Column).Sub-plot</td>
<td>Row.Column cells</td>
</tr>
</tbody>
</table>

New Prairie Press
http://newprairiepress.org/agstatconference/1993/proceedings/3
In Genstat, randomization is performed through the Randomize directive. The way that treatments are allocated to units determines the strata with which treatment effects are confounded and hence the link between design and analysis is established.

3. Procedures for generating experimental designs

In an experimental design program some simple methods allow the user access to a wide range of good designs. Firstly, many published designs may be stored in catalogues for recall. Secondly, simple constructions may be provided as procedures. Some designs could be obtained by either method so that the experimenter may choose to select a design or construct it himself. In agricultural research there has been and will continue to be need for incomplete block designs. Much of the early research on incomplete block designs concentrated on balanced and partially balanced designs with two-associate classes. These designs enable easy analysis by desk calculator but they are not necessarily efficient. Computer based analyses have freed experimenters to choose from a much wider class of design and to select more meaningful aspects of the design such as efficiency and robustness. In agricultural and biological trials resolvability, whereby groups of blocks form complete replicates, is another important design property because it enables experimenters to gain from the practical benefits of complete blocks and allows the preliminary use of a standard complete block analysis. Possibly the most successful class of incomplete block designs are the cyclic and generalised cyclic designs.

Generalised cyclic and related generators can be used to construct most popular designs. Examples of cyclic constructions include balanced incomplete block (BIB) designs, latin squares and orthogonal arrays. The designs may be generated from initial blocks stored in a catalogue or constructed by the experimenter. The properties of such designs can be determined from study of differences within the initial blocks and it is not difficult to assess which sets lead to good designs. Thus, for example, for a set of seven treatments it is easy to see that not only is the design generated from the initial block (0 1 4 2) a BIB design but also each treatment occurs as a neighbour to each other treatment in exactly one block:

\[
\begin{align*}
\text{b1} & \quad 0 & 1 & 4 & 2 \\
\text{b2} & \quad 1 & 2 & 5 & 3 \\
\text{b3} & \quad 2 & 3 & 6 & 4 \\
\text{b4} & \quad 3 & 4 & 0 & 5 \\
\text{b5} & \quad 4 & 5 & 1 & 6 \\
\text{b6} & \quad 5 & 6 & 2 & 0 \\
\text{b7} & \quad 6 & 0 & 3 & 1
\end{align*}
\]

The choice of suitable initial blocks allows a wide range of optimal and near-optimal designs to be generated easily. Tables showing the most efficient cyclic designs have been presented for example by John, Wolock and David (1972) and by Lamacraft and Hall (1982). J. A. John and co-workers
have presented a series of papers on constructing efficient
generalised-cyclic designs; much of the work is summarised
in John (1987). Cyclic designs are also useful when extra
constraints are required. Examples include cyclic designs
suitable for use as change-over designs and cyclic designs
for superimposing two sets of treatments.

An important sub-class of generalised cyclic designs
are the $\alpha$-designs for creating efficient resolvable
incomplete block designs (Patterson and Williams, 1976).
The ALPHA program provides efficient resolvable incomplete
block designs for experiments with any number of treatments
up to 500 and number of replicates up to 10. Experience
with the use of such designs over several years and many
sites has shown improvements in efficiency equivalent to
that of an extra replicate in a three-replicate trial. The
construction of a small $\alpha$-design is illustrated by an
example for 3 replicates of 18 treatments in 3 blocks of 6
units. Firstly, the numbers 1-18 are arranged in a $3 \times 6$
array. Secondly, for each replicate a vector $v$ is selected
to denote the cyclic shift to be applied to the elements of
each column. There is no loss of generality (under
randomization) if the first vector is set to zero.

<table>
<thead>
<tr>
<th>Rep 1</th>
<th>Rep 2</th>
<th>Rep 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v$</td>
<td>0 0 0 0 0 0</td>
<td>0 0 1 1 2 2</td>
</tr>
<tr>
<td>1 4 7 10 13 16</td>
<td>1 4 8 11 15 18</td>
<td>1 5 9 10 14 18</td>
</tr>
<tr>
<td>2 5 8 11 14 17</td>
<td>2 5 9 12 13 16</td>
<td>2 6 7 11 15 16</td>
</tr>
<tr>
<td>3 6 9 12 15 18</td>
<td>3 6 7 10 14 17</td>
<td>3 4 8 12 13 17</td>
</tr>
</tbody>
</table>

Within each replicate the rows form the blocks. The
properties of this design can be established from the
selected vectors $v$. For larger designs an algorithm for
finding suitable vectors for efficient $\alpha$-designs was
presented by Paterson and Patterson (1983).

Many of the generalised cyclic constructions have a
quasi-factorial nature. Thus for example, the six
replicates of a balanced lattice design for 25 treatments in
5 blocks of 5 units may be generated through representing
the treatments as combinations of two 5 level pseudo-factors
and then generating modulo 5 from two initial treatments per
replicate.

\begin{align*}
(1 \ 0) & \ (0 \ 1) ; \ (0 \ 1) \ (1 \ 1) ; \ (0 \ 1) \ (2 \ 1) ; \\
(0 \ 1) & \ (3 \ 1) ; \ (0 \ 1) \ (4 \ 1) ; \ (0 \ 1) \ (1 \ 0) \\
\end{align*}

Thus the second replicate formed by $(1 \ 0) \ (1 \ 1)$ becomes

<table>
<thead>
<tr>
<th>b1</th>
<th>b2</th>
<th>b3</th>
<th>b4</th>
<th>b5</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>01</td>
<td>02</td>
<td>03</td>
<td>04</td>
</tr>
<tr>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
</tr>
<tr>
<td>33</td>
<td>34</td>
<td>35</td>
<td>36</td>
<td>37</td>
</tr>
<tr>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
</tr>
</tbody>
</table>

Monod and Bailey (1992) describe how pseudo-factors may be
used to construct efficient designs and to aid the analysis.
Methods for the construction of factorial designs are
described in the next section.
4. Factorial Designs

As with single factor designs, a few tools are required whereby a wide range of practically useful experimental designs may be generated. We concentrate on orthogonal fractional and confounded designs and on main-effects designs. Generalised cyclic type generators again construct a wide range of useful designs. A powerful and compact tool for generating orthogonal fractional and confounded designs is the design-key (see e.g. Patterson and Bailey (1978), Bailey (1978)). This tool not only provides a compact expression for the treatments but also, for symmetrical factorial designs, it provides a similar expression for the defining contrasts. It works by identifying an effective symmetry between the effects (or contrasts) among the units and those among the treatments. If necessary pseudo-factors are used. Equating treatment effects and unit effects determines both defining contrasts and treatments.

Example: we construct a single replicate $2^5$ factorial in 4 blocks of 8 units with ACE, BDE, ABCD confounded. Represent blocks by 2 two-level pseudo-factors $P_1$, $P_2$, and units within blocks by 3 two-level pseudo-factors $P_3$, $P_4$, $P_5$. Identify ACE with $P_1$, BDE with $P_2$ (and hence ABCD=ACE.BDE with $P_1P_2$). Main effects $C$, $D$, $E$ can be identified with $P_3$, $P_4$, $P_5$ respectively. Each treatment effect now has an equivalent unit effect, e.g. A equates to $P_1P_3P_5$, ABC equates to $P_1P_2P_4$. Any treatment effect identified with unit effects $P_1$, $P_2$, or $P_3$. The identity can be summarised by writing the treatment effects associated with each main effect as the columns of matrix $L$.

\[
\begin{array}{ccccc}
P_1 & P_2 & P_3 & P_4 & P_5 \\
A & 1 & 0 & 0 & 0 & 0 \\
B & 0 & 1 & 0 & 0 & 0 \\
C & 1 & 0 & 1 & 0 & 0 \\
D & 0 & 1 & 0 & 1 & 0 \\
E & 1 & 1 & 0 & 0 & 1 \\
\end{array}
\]

Note that the first two columns, associated with blocks, contain the defining contrast generators. Inversion of $L$ (working in fields of order of 2) yields the matrix $K=L^{-1}$.

\[
\begin{array}{ccccc}
P_1 & P_2 & P_3 & P_4 & P_5 \\
A & 1 & 0 & 0 & 0 \\
B & 0 & 1 & 0 & 0 \\
C & 1 & 0 & 1 & 0 \\
D & 0 & 1 & 0 & 1 \\
E & 1 & 1 & 0 & 1 \\
\end{array}
\]

The rows associated with $P_3$, $P_4$ and $P_5$ identify the treatment generators for the principal block. When combined with the rows for $P_1$, $P_2$ the whole design is generated. The matrix $K$ is usually referred to as the ‘key matrix’. Observe that the last three rows of $K$ form the key matrix for generating a fractional $2^5-2$ design with defining contrasts ACE, BDE, ABCD. The matrix $K$ could also be used to generate the
fractional design if it is recognised that $P_1$ and $P_2$ correspond to the fraction. For fractional $p^{n-m}$ designs, $p$ prime, $m$ unit factors can represent the fraction; any treatment effect which is identified with an effect among these factors is a defining contrast.

The idea of defining contrast generators is not new and the relationship between $K$ and $L$ is clearly similar to that between an error-correcting code matrix and a parity check matrix (see e.g. Pless, 1989). For standard designs, the neatness of the design-key method is that it allows a simple expression of the defining contrasts and treatment generators. For non-standard designs, however, it can be more effective so that, say, a single replicate $2^5$ design in a design with 4 rows and 4 columns with 2 units per cell still requires only a 5 x 5 matrix $L$ to be defined and inverted for solution. Whether $L$ is derived from $K$ or $K$ from $L$ depends on the problem and the individual; mostly, the defining contrasts are chosen and the treatments then derived. Das and Giri (1986, p120) directly construct the matrices $K$, a procedure which can be advantageous for difficult problems.

Tables of defining contrasts for standard symmetric $p^{n-m}$ factorial designs e.g. minimum aberration designs (Franklin, 1984, and Chen, Sun and Wu, 1993) are often most compactly expressed in the form of a matrix $L$ such that $L = (I C)^T$. (Note the first two columns of $L$ above.) This is sometimes known as reduced echelon form. It is easy to show the treatments in the principal block can be generated from the rows of $K = (pJ - C I)$ where $J$ has all elements equal to unity. The design key method therefore adapts well to such standard designs. It also adapts well to more specific problems such as finding all solutions, if any, for a $2^6-2$ design in which all main effects and all two-factor interactions involving factor A are estimable. Franklin (1985) outlines a search routine for a suitable design generated from a matrix $L$ in reduced echelon form.

Partially confounded designs and asymmetric factorials are generally less convenient to handle. The former usually require a different key matrix for each replicate. For asymmetric factorials the simple matrix inversion rule usually fails but experience in handling key matrices leads the user quickly to good solutions to tricky problems. For example, the following key matrix may be used to construct a balanced partially confounded $3.2^2$ factorial in 3 replicates, each replicate consisting of two blocks of six units.

$$
\begin{array}{ccc}
A & B & C \\
P_1 & 1 & 0 & 0 \\
P_2 & 0 & 1 & 0 \\
P_3 & 1 & 0 & 1 \\
P_4 & 0 & 1 & 1 \\
\end{array}
$$
Here unit factors $P_1, P_2$ correspond to replicates and blocks respectively and $P_3, P_4$ are 3 and 2-level factors corresponding to the 6 units within a block. The design is equivalent to that in plan 6.9 of Cochran and Cox (1957).

Orthogonal main-effect designs derived from $p^{n-m}$ factorial designs are readily handled by the design-key method. However, other main effect plans are not. Thus, for example, it is unsuited to generating the main-effect plan for 11 2-level factors in 12 units. However, many such plans are either derived directly from Hadamard matrices or orthogonal arrays derived from simple cyclic generators applied to (generalised) Hadamard matrices (Wang and Wu, 1991).

5. Some simple combinatorial operations

An experimental design can be viewed as a two-way array with rows corresponding to experimental units and columns to block and treatment factors. It is important that each unit be uniquely identified by the levels of the blocking factor. The design may then be perceived as a function $t=f(b)$ defining for each experimental unit the treatment applied. The ability to manipulate structures allows the experimenter to modify the design in various ways. Three basic operations provide much versatility to the construction process:

(i) manipulating factor levels,
(ii) manipulating factors and
(iii) forming one design as a subset of another or combining two designs to make a third.

Operations of the first type include permuting the levels, either controlled or at random, and ‘folding’ the levels of a factor so that, for example, levels 0,1,2,3 become 0,1,2,1. Of the second type, operations include the formation of pseudo-factors where one factor is replaced by more than one or vice-versa. Also, block and treatment factors may be interchanged so that, for example, the single replicate design $t=f(b)$ has dual design $b=f^{-1}(t)$. Central composite designs are an example of the third operation as are designs in which extra control treatments are added to factorial designs. They may be formed by combining separate designs or by building the design one component at a time. Other such operations allow the super-imposition of one set of treatments on another and the addition of sub-plots to existing designs.

A few illustrations serve to illustrate how such combinatorial operations may be used in constructing designs. (a) In section 3 we noted how a lattice design could be constructed through use of pseudo-factors. (b) Deletion of treatment 18 in the alpha design of section 3 yields an efficient design for 17 treatments in blocks of size 5 or 6. (Treatments 16 and 17 could also be deleted...
with a minimum block size of 5 being retained). (c) The confounded 3.2² design of section 4 can be derived from a balanced 4.2² design by deletion of one level of the first factor (cf Plans 6.9, 6.13, Cochran and Cox (1957)). (d) The formation of fractional 2⁶3² factorials by the method of conjoining fractions (Connor and Young, 1959) essentially involves the union of direct products of fraction 2-level and 3-level factorials. (In practice many of these designs are more simply generated through design-key procedures.)

Two or more of these construction techniques may be used in one design. Thus, for example, a central composite design has three distinct components - the basic factorial, the axial points and the centre points. A design could be constructed from these parts followed by, say, addition or deletion of some centre points until a design with the desired characteristics is achieved.

6. Randomization

Randomization can be viewed as a method of forming one design from another but it is usually best treated as a separate component of the process for constructing a design. Most of the designs commonly used in agricultural and biological research use orthogonal block structures and we noted in section 2 how such structures impose restrictions on the randomization and partition the analysis of variance. Nelder (1965) describes a general theory for the randomization analysis of designs with orthogonal block structures.

In most designs any permutation of factor levels is permitted. Occasionally, however, constraints are placed on the randomization of the levels of one or more factors. The most extreme constraint occurs when the randomization is totally suppressed for some factors. This may occur, for example, in changeover or repeated measures designs where randomization of treatment sequences may cause a design to become unbalanced for carry-over effects. Similarly sequences which are robust to simple trends lose that property under randomization. Such extreme restrictions often lead to problems of selecting valid forms of analysis. Sometimes, however, it is possible to apply combinatorial constraints to the randomization and yet not invalidate the standard analysis. The restrictions are chosen so as to avoid the 'worst' randomizations. Examples of such restricted randomization include the allocation of 2-level factorials to blocks of 8 units by Grundy and Healy (1950) and the allocation of k treatments to blocks of sk units by Sutter, Zyskind and Kempthorne (1963). In small trials where degrees of freedom are scarce such restricted randomization can be superior to the use of blocks. Constraints on randomization can also be applied in the presence of covariates. Here, criteria may be defined for the rejection of randomizations with a low covariance efficiency; repeated randomizations can be used to establish that the criteria are not too restrictive (Cox, 1982).
7. Output of designs

The simplest output for a design is as a list with rows corresponding to units and columns containing the levels of associated unit and treatment factors. This list may then be used as the input to routines for dummy analysis of variance or for the production of recording forms and experimental plans. It may also be stored for use in the subsequent analysis of data. Normally the most useful ordering of the units is that determined by the lexicographical sequence of the factors in the block structure. However, when this sequence does not tally with that required by the experimenter, it may be necessary to give each unit a sequence number and an identifier.

Block structures are useful for producing one and two-dimensional plans for they define the hierarchy of the units and the relationships that may exist between the block shapes. In this respect it is convenient to think of the design itself as being a factor, with one level, within which the units are nested. (It corresponds to the correction factor!) Thus a 4 x 4 Latin square with 4 subplots per cell may have a layout defined by

<table>
<thead>
<tr>
<th>rows</th>
<th>columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>(design)</td>
<td>4</td>
</tr>
<tr>
<td>rows</td>
<td>1</td>
</tr>
<tr>
<td>column</td>
<td>4</td>
</tr>
<tr>
<td>subplot</td>
<td>1</td>
</tr>
</tbody>
</table>

Each factor can have its own symbol for printing borders and common borders are printed with the symbol of the first named factor in the block structure. Printing plans displaying selected information on factor treatments can help experimenters spot deficiencies of which they are unaware.

An alternative form of printing also of great practical benefit to experimenters is one in which the unit or units corresponding to each treatment, or selected factor combinations, are listed. Besides prompting warnings of potential problems, it enables the experimenter to quickly identify all units having selected treatments. This can be convenient both for practical supervision and data handling.

8. Design construction in Genstat

We now briefly outline how some of the design requirements outlined above are satisfied within Genstat. The GENERATE directive provides a simple way of constructing arrays of block and treatment factors in a systematic order. It can be used directly to construct factorial designs in complete blocks or split-plot designs. It forms the basis of various procedures for constructing other designs so that procedure AKEY is used to generate factorial designs by the design key method while AFCYCLIC and AFALPHA are used to
generate cyclic and $\alpha$-designs respectively. The RESTRICT directive, in conjunction with the EQUATE directive can be used to build up a design component at a time or to select subsets of larger designs. The roles of the factors in a design $D_1$, say, may be changed by defining a new design $D_2$ based on the same factors as $D_1$ but with different block and treatment structures.

Designs with orthogonal block structure can be randomized through directive RANDOMIZE. The randomization of the levels of selected block factors can be suppressed. Constrained randomization in the presence of covariates can be achieved through a sequence of operations, randomization - dummy analysis of covariance - randomization etc., using the covariance efficiency factors to define acceptable allocations. Some but not all of the procedures for printing designs are available.

The ability to obtain an effective design is greatly enhanced by interactive working particularly through a sequence of design - analysis - design.... The dummy analysis of variance and covariance facility available in Genstat is helpful for establishing that a design is analysable and has the properties required by the experimenter. However, it is possibly more useful for the experimenter to create a dummy variable for analysis. This variable may simulate the type of variation the experimenter expects or it may be a mundane listing of, say, plot locations which, on analysis, could expose some unfortunate aspect of a randomization.

9. Summary

Simple procedures, such as those we have described, for constructing a wide range of useful designs could be easily added to most statistical analysis packages. The easy access to good designs and to improved randomizations within a package providing analysis of these designs should increase their use by experimenters. Awareness of the advantages of good designs would lead to a gradual extension in the application of effective designs.
References


