Automating the analysis of variance of orthogonal designs

Heiko Großmann [*]

School of Mathematical Sciences, Queen Mary University of London, Mile End Road, London E1 4NS, UK

Abstract

A new algorithm is presented which for a wide class of designs is capable of deducing the appropriate analysis of variance from the design only. As a consequence the use of a model formula for specifying the analysis becomes dispensable. This unique feature distinguishes the current approach from other existing procedures. A user-friendly implementation is described and the use of the program is illustrated with several examples.

Key words: Analysis of variance, Orthogonal block structure, Hasse diagram, Algorithm
1991 MSC: 62 J10

1 Introduction

This paper is about a computer program for the analysis of variance (anova). Even before finishing the reading of the first sentence the reader may wonder if another such program is needed. In the paper I will try to answer this question in the affirmative by explaining how the proposed *AutomaticAnova* package can simplify the analysis of complex anova models with complicated blocking structures and factors having random and fixed effects in a way, I believe, that no other existing package can. Of course, this statement needs to be taken with a pinch of salt, because although being very general the theory underlying the program has its limitations and, essentially, only applies to orthogonal designs as defined by Bailey (2008). Also, in order to prevent later confusion, it seems to be appropriate to point out from the beginning that,

[*] Corresponding author.
Email address: h.grossmann@qmul.ac.uk (Heiko Großmann).

despite being a familiar term, the name ‘orthogonal design’ means different things to different people and therefore being clear about the definition used in this paper is important.

The *AutomaticAnova* package originated from two sources. One was the teaching of a module *Design of Experiments* originally designed by R. A. Bailey, the material of which is now available in Bailey (2008). The other was an interdisciplinary collaboration with biologists on the personality of bumblebees reported in Muller et al. (2010). From teaching the module it became clear that the theory had some algorithmic content which had not been recognized before and which could lead to the anova being automated. The joint work with the biologists on the other hand revealed that a software implementation of Bailey’s theory could tremendously reduce the time needed for providing consultancy. Moreover, in order to be useful to non-statisticians, usage of the software had to be very simple and intuitive. The final product appears to meet both criteria.

At this point it seems to be appropriate to give the reader an idea about what ‘automating the analysis of variance’ means. In short, this phrase refers to the *AutomaticAnova* package’s capability to *infer* an appropriate model from a design provided in the form of a spreadsheet and to carry out the corresponding analysis *without the need to specify a model formula.* From a practical point of view the fact that the user does not have to specify the model appears to be crucial, since from experience it seems that non-statisticians usually find it hard to understand and to correctly apply the operators, such as nesting and crossing, which are commonly offered by software for defining anova models. It is also worthwhile to note that the *AutomaticAnova* package’s algorithm for deriving the model from a design is not based on a finite collection of predefined designs or models, but works for any orthogonal design.

The sections that follow provide a brief account of the theory on which the *AutomaticAnova* package is based, present the algorithm for automating the anova and give some information on the implementation. In addition, the practical use of the package is explained and illustrated with several examples. The paper concludes with a discussion of limitations and further extensions.

2 Bailey’s theory for orthogonal designs

The analysis of variance is one of the most versatile and popular statistical techniques in common use. Although the principles on which the anova is based are well understood, there exist different perspectives on the method. In one, the anova is regarded as an instance of the general linear model and the analysis is considered from a regression point of view based on projection matrices (e.g., Christensen, 1987). Other presentations focus on model equations and corresponding decompositions of sums of squares and degrees of freedom (Montgomery, 1991; Sahai and Ageel, 2000). A third approach puts more emphasis on randomization ideas and clearly distinguishes the structure of the observational or experimental units from the structure of the treatments (Nelder, 1965a,b), hence the model formula plays a secondary role. The practical differences between the approaches for mixed anova models with fixed and random effects become apparent when one looks at software packages such as SAS (see, e.g., Littell et al., 2006), which is close in spirit to the first two approaches, and GenStat, which implements the third approach. Interestingly, it is not uncommon to find that proponents of different anova 'schools' have difficulties understanding each other. A thorough discussion of these matters is beyond the scope of this paper, but more information can be found, for example, in the discussion papers by Speed (1987) and Gelman (2005).

In this paper, the focus is on a version of the third approach presented in the monograph by Bailey (2008), which generalizes the seminal work of Nelder (1965a,b). The brief summary of the theory below is necessary for understanding the material in Section 3. Related ideas have been presented by Houtman and Speed (1983), Tjur (1984) and Bailey (1981, 1996). Readers who are familiar with Bailey's approach to the anova may skip the rest of this section and only use it as a reference later on.

2.1 Definitions and notation

In what follows, for the most part I adopt the notation in Bailey (2008) to facilitate comparisons of the material in this paper with the more comprehensive account in Bailey's book. Where a modified notation is used this is motivated by trying to make some aspects of the theory more explicit.

The theory distinguishes the set Ω of size N which represents the observational units from the set $\mathcal{T}$ of treatments which has size t. A plot (or block) factor F is a function from Ω to a finite set of n_F levels and similarly a treatment factor G is a function from $\mathcal{T}$ to a finite set of n_G levels. In accordance with Nelder (1965a), plot factors reflect the structure, natural or otherwise, of the units, such as arrangements into blocks, ignoring the treatments, whereas treatment factors indicate what is done to the units. For simplicity of presentation we assume that the elements of Ω are the integers $1, \ldots, N$ and that for every factor H on Ω or $\mathcal{T}$ with n_H levels these are also represented by integers $1, \ldots, n_H$.

Each plot factor F gives rise to a partition of Ω into the F-classes $F[[i]] = \{\omega \in \Omega : F(\omega) = i\}$ for $i = 1, \ldots, n_F$, and likewise a treatment factor G

introduces a partition of $\mathcal{T}$ into G-classes $G[[j]] = \{a \in \mathcal{T} : G(a) = j\}$, where $j = 1, \ldots, n_G$. Thus each factor H with n_H levels can be identified with a set of n_H sets, which are the H-classes. Bailey (2008, p. 169) defines the classes associated with a factor in a slightly different way, but both definitions lead to the same partitions.

There are two special factors each of which can be defined on Ω or $\mathcal{T}$. The universal factor U has only a single level and hence also a single U-class equal to Ω or $\mathcal{T}$ depending on whether U is regarded as a factor on the set of observational units or the set of treatments. The equality factor E has as many levels and hence E-classes as there are elements in Ω or $\mathcal{T}$. For E defined on Ω these classes are the singletons $\{\omega\}$, $\omega \in \Omega$, whereas for E defined on $\mathcal{T}$ the E-classes are all sets $\{a\}$, where $a \in \mathcal{T}$. Since it should always be clear from the context on which set the factor U (or E) is defined I will use the same symbol U (or E) for the factors on Ω and $\mathcal{T}$.

By identifying each plot factor F with the corresponding partition $\{F[[i]] : i = 1, \ldots, n_F\}$, any given collection of plot factors can be partially ordered in a simple way. To this end, let $\mathcal{F}$ be a finite set of plot factors. Two factors $F, G \in \mathcal{F}$ are equivalent, denoted by $F \equiv G$, if they have the same classes, that is if every F-class is equal to some G-class and vice versa. Otherwise they are called inequivalent. Moreover, F is said to be finer than G (or G to be coarser than F) if the factors are inequivalent and if for every $i = 1, \ldots, n_F$ there exists a $j \in \{1, \ldots, n_G\}$ such that $F[[i]] \subseteq G[[j]]$. This is denoted by $F \prec G$ or $G \succ F$. For inequivalent factors $F, G \in \mathcal{F}$, in words $F \prec G$ means that whenever two units in Ω have the same level of F then they also have the same level of G. Finally, a factor $F \in \mathcal{F}$ is finer than or equivalent to $G \in \mathcal{F}$, which is denoted by $F \preceq G$ (or $G \succeq F$), if $F \prec G$ or $F \equiv G$. Any set $\mathcal{F}$ of plot factors can then be partially ordered in terms of the relations $\prec$ or $\preceq$. Likewise, any set of treatment factors $\mathcal{G}$ on $\mathcal{T}$ can be partially ordered in terms of similarly defined relations $\prec$ or $\preceq$. Clearly, the finer relation $\prec$ is irreflexive and hence it is a strict partial order.

In addition to being able to separately define partial orders for sets of plot and treatment factors, new factors can be created from old ones by means of two binary operators $\wedge$ and $\vee$. Since the operators are defined in the same way for plot and treatment factors, I only present the details for plot factors. Suppose F and G are two plot factors. Then the infimum of F and G, which is denoted by $F \wedge G$, is defined as the coarsest plot factor which is finer than or equivalent to F and finer than or equivalent to G. More formally, $F \wedge G$ is the infimum of F and G if (a) $F \wedge G \preceq F$ and $F \wedge G \preceq G$ and (b) for every plot factor H with $H \preceq F$ and $H \preceq G$ it follows that $H \preceq F \wedge G$. The supremum of F and G, represented by $F \vee G$, is defined as the finest plot factor which is coarser than or equivalent to F and coarser than or equivalent to G. In formal terms, $F \vee G$ is the supremum of F and G if (a) $F \preceq F \vee G$

and $G \preceq F \vee G$ and (b) for every plot factor H with $F \preceq H$ and $G \preceq H$ it follows that $F \vee G \preceq H$. Both the infimum $F \wedge G$ and the supremum $F \vee G$ are unique up to equivalence.

Given two factors F and G on the same set, finding the infimum is always easy, whereas finding the supremum is more complicated and this will be considered in more detail in Section 3. Both $\wedge$ and $\vee$ are commutative operators. Also it is worth noting that the two operators are associative, that is $(F \wedge G) \wedge H = F \wedge (G \wedge H)$ and $(F \vee G) \vee H = F \vee (G \vee H)$, which implies that infima and suprema of three or more factors are also well-defined.

With these definitions in place one can move on to define what it means for two factors F and G on the same set (Ω or $\mathcal{T}$) to be orthogonal. Every class $(F \vee G)[[k]]$, $k = 1, \ldots, n_{F \vee G}$, of the supremum $F \vee G$ is the union of a certain number of F-classes. At the same time $(F \vee G)[[k]]$ is also the union of some of the G-classes. For every k suppose that $(F \vee G)[[k]] = F[[i_{k,1}]] \cup \cdots \cup F[[i_{k,u_k}]]$ where $\{i_{k,1}, \ldots, i_{k,u_k}\} \subseteq \{1, \ldots, n_F\}$ and $(F \vee G)[[k]] = G[[j_{k,1}]] \cup \cdots \cup G[[j_{k,v_k}]]$ where $\{j_{k,1}, \ldots, j_{k,v_k}\} \subseteq \{1, \ldots, n_G\}$. Then F and G are defined to be orthogonal if for every $k = 1, \ldots, n_{F \vee G}$ the following two conditions are satisfied: (a) $F[[i_{k,r}]] \cap G[[j_{k,s}]] \neq \emptyset$ for $r = 1, \ldots, u_k$ and $s = 1, \ldots, v_k$ and (b) there exists a constant c_k such that for $r = 1, \ldots, u_k$ and $s = 1, \ldots, v_k$ it holds that the size of $F[[i_{k,r}]] \cap G[[j_{k,s}]]$ divided by the product of the sizes of $F[[i_{k,r}]]$ and $G[[j_{k,s}]]$ is equal to c_k. Bailey (2008, p. 179) defines the orthogonality of two factors in a different way, but her Theorem 10.5 shows that both definitions are equivalent. She also notes that the characterization of orthogonality given here is more useful for checking if two factors are orthogonal.

We are now in a position to define orthogonal designs in the sense of Bailey (2008, p. 198). Such a design is characterized by twelve properties some of which can be checked very quickly while verifying others can take some time. Let $\mathcal{F}$ be a set of mutually inequivalent plot factors on Ω and $\mathcal{G}$ be a set of mutually inequivalent treatment factors defined on $\mathcal{T}$. In other words, any two factors in $\mathcal{F}$ are not equivalent and the same applies to any pair of factors in $\mathcal{G}$. Further suppose that the treatments are allocated to the observational units according to a design function T which is defined on Ω and takes values in $\mathcal{T}$. Thus $T(\omega)$ is the treatment assigned to observational unit ω. It is worth noting that for every factor $G \in \mathcal{G}$ the composition $G \circ T$, where $(G \circ T)(\omega) = G(T(\omega))$, is also a factor on Ω. By definition the triple $(\mathcal{F}, \mathcal{G}, T)$ is then an orthogonal design if

(i) $\mathcal{F}$ is an orthogonal plot structure (Bailey, 2008, p. 194), that is
  (a) if $F \in \mathcal{F}$ then all its F-classes have the same size,
  (b) $U \in \mathcal{F}$,
  (c) $E \in \mathcal{F}$,
  (d) if $F \in \mathcal{F}$ and $G \in \mathcal{F}$ then $F \vee G \in \mathcal{F}$,

(e) if $F \in \mathcal{F}$ and $G \in \mathcal{F}$ then $F \wedge G \in \mathcal{F}$,

(f) if $F \in \mathcal{F}$ and $G \in \mathcal{F}$ then F is orthogonal to G,

(ii) $\mathcal{G}$ is an orthogonal treatment structure (Bailey, 2008, p. 190), that is

(a) $U \in \mathcal{G}$,

(b) if $F \in \mathcal{G}$ and $G \in \mathcal{G}$ then $F \vee G \in \mathcal{G}$,

(c) if $F \in \mathcal{G}$ and $G \in \mathcal{G}$ then F is orthogonal to G,

(iii) the function T is such that

(a) if $F \in \mathcal{G}$ and $G \in \mathcal{G}$ then $F \circ T$ and $G \circ T$ are orthogonal on Ω,

(b) if $F \in \mathcal{F}$ and $G \in \mathcal{G}$ then F and $G \circ T$ are orthogonal on Ω,

(c) if $F \in \mathcal{F}$ and $G \in \mathcal{G}$ then $F \vee (G \circ T)$ is equal to $H \circ T$ for some $H \in \mathcal{G}$.

The following comments about the definition may be helpful. Bailey (2008) calls a factor with property (i)(a) uniform, but more commonly such a factor is referred to as a balanced factor in the experimental design literature. If F and G are both in $\mathcal{F}$ then the common size of the F-classes can be different from the common size of the G-classes. Part (iii) of the definition explicitly mentions the design function T in (a), (b) and (c) which is slightly different from the exposition in Bailey (2008, p. 198), but hopefully adds some clarity.

Trying to provide a motivation for each of the twelve properties in the definition of an orthogonal design would amount to repeating large portions of Bailey (2008, Chapter 10) and the interested reader is referred to the original source. However, it is worth noting that the class of orthogonal designs defined here includes the simple orthogonal block structures of Nelder (1965a,b), but also some other types of design (see, e.g., Bailey, 1996, p. 60) and hence is more general. The main reason for including the admittedly lengthy definition of orthogonal designs here is to introduce the designs in a form that is suitable for describing the *AutomaticAnova* algorithm is Section 3.

2.2 Orthogonal designs, Hasse diagrams, and the anova table

The main advantage of considering orthogonal designs $(\mathcal{F}, \mathcal{G}, T)$ is that they allow a particularly simple analysis. Moreover, for these designs the calculation of degrees of freedom and sums of squares can be easily performed by using Hasse diagrams, one for the set of plot factors $\mathcal{F}$ and one for the treatment factors in $\mathcal{G}$, which represent the partial order $\prec$. Two examples are shown in Figure 1.

The Hasse diagram for the plot factors $\mathcal{F}$ is a directed graph, in which the elements of $\mathcal{F}$ are the vertices and where there is an edge with endpoints $F, G \in \mathcal{F}$ if and only if (a) $F \prec G$ and (b) there exists no other $H \in \mathcal{F}$ with $F \prec H \prec G$. In this case the vertex for F is drawn below the vertex for

G. Because of this convention no arrows are needed when a Hasse diagram is drawn. If F and G are inequivalent and neither $F \prec G$ nor $G \prec F$, where possible the vertices for F and G are displayed beside each other, so that none of the vertices for the two factors is above or below the other. Next to each vertex for a factor $F \in \mathcal{F}$ we write the number n_F of its levels. Because $\prec$ is only a partial order, the Hasse diagram for $\mathcal{F}$ will usually not be a complete graph, that is not all vertices will be connected by edges.

Part (i) of the definition of an orthogonal design implies that the vertices for U and E on Ω will always be at the top and bottom of the Hasse diagram for the plot factors, respectively. Moreover, part (i) of the definition also ensures that the vertex for the supremum $F \vee G$ of two inequivalent factors $F, G \in \mathcal{F}$ where neither $F \prec G$ nor $G \prec F$ is always above and connected to both F and G, though not necessarily via single edges. Likewise, the vertex for the infimum $F \wedge G$ of two such factors $F, G \in \mathcal{F}$ is always below and connected to the vertices for F and G.

It should be noted that for any two inequivalent plot factors $F, G \in \mathcal{F}$ of an orthogonal design both the infimum $F \wedge G$ and the supremum $F \vee G$ are also elements of $\mathcal{F}$, but often the corresponding factors may have different names or labels. For example, if $F \prec G$, then $F \wedge G = F$ and $F \vee G = G$ so that, if in addition neither F nor G is equivalent to either U or E on Ω, the set $\mathcal{F} = \{E, F, G, U\}$ satisfies the conditions (i)(b)–(i)(e) in the definition of an orthogonal design, but $\mathcal{F}$ does not contain additional factors called $F \wedge G$ or $F \vee G$ and there will be no extra vertices for these factors in the Hasse diagram for plot factors.

The Hasse diagram for the set of treatment factors $\mathcal{G}$ of an orthogonal design $(\mathcal{F}, \mathcal{G}, T)$ is drawn in exactly the same way as the diagram for the plot factors $\mathcal{F}$. Because of the weaker requirements in part (ii) of the definition there are, however, some differences. First, the diagram for the treatment factors always includes a vertex for the universal factor U on $\mathcal{T}$ at the top, but not necessarily a vertex for the equality factor E on $\mathcal{T}$. Secondly, for every pair of inequivalent treatment factors $F, G \in \mathcal{G}$ which are such that neither $F \prec G$ nor $G \prec F$ the Hasse diagram always includes a vertex for the supremum $F \vee G$, but not necessarily a vertex for the infimum $F \wedge G$. Whether the equality factor and all or selected infima form part of the diagram depends on whether they are elements of $\mathcal{G}$. Put differently, whether these factors appear in the diagram depends only on which factors are included in $\mathcal{G}$, which is decided before the drawing stage. How this is done will be considered further in Section 3. However, if $\mathcal{G}$ contains the equality factor or the infimum of two factors, then in the Hasse diagram for treatments factors these are represented exactly as in the diagram for plot factors.

Theorems 10.6 and 10.7 in Bailey (2008, pp. 182–184) provide the basis of a simple procedure for using the Hasse diagrams to calculate degrees of freedom and corresponding sums of squares for the plot and treatment factors of an orthogonal design. Since the procedure is the same for plot and treatment factors, here I only describe the details for plot factors.

For every plot factor F the degrees of freedom are denoted by d_F and defined as the dimension of a corresponding subspace W_F of the vector space $\mathbb{R}^N$ (Bailey, 2008, p. 183). Theorem 10.6 then shows that for every $F \in \mathcal{F}$ the degrees of freedom d_F are equal to the number of levels n_F minus the sum of the degrees of freedom of all plot factors $G \in \mathcal{F}$ that are coarser than F: in symbols $d_F = n_F - \sum_{G \succ F} d_G$. Since on Ω no factor is coarser than the universal factor U, it follows that $d_U = n_U = 1$. This result translates into a graphical procedure on the Hasse diagram for plot factors as follows: Start with the universal factor U at the top of the diagram and annotate the diagram by writing down the degrees of freedom $d_U = 1$ next to the number of levels $n_U = 1$, preferably using a color or font which is different from the one used for the numbers of levels. Subsequently, move down the diagram along the edges. At each vertex F, subtract the degrees of freedom of all factors $G \succ F$ whose vertices are above and connected to F from the number of levels n_F to obtain d_F and annotate the diagram with the degrees of freedom for F. This process terminates with the equality factor E at the bottom of the diagram.

Theorem 10.7 leads to a similar procedure for calculating the sum of squares $\mathrm{SS}(F)$ associated with each plot factor F. In this, for every $F \in \mathcal{F}$ the number of levels n_F in the Hasse diagram is replaced with the crude sum of squares $\mathrm{CSS}(F)$, which is defined as the squared Euclidean length of the orthogonal projection of the data vector onto another subspace V_F of $\mathbb{R}^N$ (Bailey, 2008, p. 184), and the degrees of freedom d_F are replaced with $\mathrm{SS}(F)$, which is defined as the squared Euclidean length of the orthogonal projection of the data vector onto the vector space W_F mentioned before. With these modifications the calculation of the sums of squares proceeds as before by working down the Hasse diagram from U, since, as follows from Theorem 10.7, $\mathrm{SS}(F) = \mathrm{CSS}(F) - \sum_{G \succ F} \mathrm{SS}(G)$ and, in particular, $\mathrm{SS}(U) = \mathrm{CSS}(U)$.

For the treatment factors in $\mathcal{G}$ degrees of freedom and sums of squares are calculated in the same way by using the Hasse diagram for treatment factors. Since it is not immediately obvious from Bailey's presentation, it should be noted that, at least when sums of squares are calculated, for every $G \in \mathcal{G}$ the vector spaces $V_{G \circ T}$ and $W_{G \circ T}$ are used, where T is the design function of the orthogonal design $(\mathcal{F}, \mathcal{G}, T)$. Also because the Hasse diagram for the treatment factors does not necessarily contain the equality factor E on $\mathcal{T}$ nor the infimum of $F \wedge G$ for every pair of factors $F, G \in \mathcal{G}$ the process of calculating degrees of freedom or sums of squares does not necessarily terminate at the vertex for E but can end at two or more vertices corresponding to mutually inequivalent factors other than E. The following example shows how the procedure for

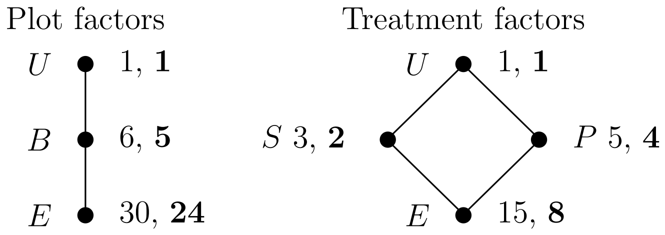


Fig. 1. Hasse diagrams for plot and treatment factors in Example 1

calculating degrees of freedom works in practice.

Example 1 In an experiment reported by Bailey (2008, p. 145) thirty apple trees were grouped into six blocks of size five. Each of three spray treatments was applied to two whole blocks chosen at random. Within each block each of five methods of pruning was used on a single randomly selected tree.

Here the set Ω of observational units contains the integers $1, \ldots, 30$ representing the trees. The arrangement of the trees into blocks is reflected by a plot factor B where for every $\omega \in \Omega$ the value $B(\omega)$ is the number of the block containing the tree ω. Other trivial plot factors are the universal factor U and the equality factor E which distinguishes the different trees. Thus for this experiment $\mathcal{F} = \{E, B, U\}$ is a natural set of plot factors and $E \prec B \prec U$.

The set of treatments $\mathcal{T}$ contains the fifteen combinations of spray treatments and pruning methods. For each $a \in \mathcal{T}$ we can consider the treatment factor S, where $S(a)$ is the spray component of a, and the factor P, where $P(a)$ is the pruning component of a. On $\mathcal{T}$ the equality factor E distinguishes between the treatments and it can be verified that $E = S \wedge P$. Furthermore, for the universal factor U on $\mathcal{T}$ it can be seen that $U = S \vee P$. A possible choice for the set of treatment factors is then $\mathcal{G} = \{E, S, P, U\}$. For these factors it holds that $E \prec S \prec U$ and $E \prec P \prec U$, but neither $S \prec P$ nor $P \prec S$.

For every $\omega \in \Omega$ the design function T gives the treatment combination a_ω in $\mathcal{T}$ that was applied to tree ω, that is $T(\omega) = a_\omega$. It can then be shown that the triple $(\mathcal{F}, \mathcal{G}, T)$ is an orthogonal design.

Figure 1 presents the Hasse diagrams for plot and treatment factors, where the degrees of freedom are shown in bold. From the diagram for the plot factors it can be easily seen that, for example, the degrees of freedom for E on Ω are equal to the number of trees minus the degrees of freedom for all coarser factors above E, that is $d_E = n_E - d_U - d_B = 30 - 1 - 5 = 24$. As was explained before, the degrees of freedom for other factors are obtained by the same method. Apart from the numbers, the Hasse diagrams for the sums of squares (not shown here) are identical to those for the degrees of freedom. □

Bailey (2008, p. 193) assumes that the plot factors have random effects and

Table 1
Portion of the anova table for a stratum $F \in \mathcal{F}$ containing factors $G_1, \ldots, G_k \in \mathcal{G}$

Stratum	Source	SS	DF	MS	F
F	G_1	$\mathrm{SS}(G_1)$	d_{G_1}	$\mathrm{MS}(G_1) = \frac{\mathrm{SS}(G_1)}{d_{G_1}}$	$\frac{\mathrm{MS}(G_1)}{\mathrm{MS(res)}}$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	G_k	$\mathrm{SS}(G_k)$	d_{G_k}	$\mathrm{MS}(G_k) = \frac{\mathrm{SS}(G_k)}{d_{G_k}}$	$\frac{\mathrm{MS}(G_k)}{\mathrm{MS(res)}}$
	residual	SS(res)	d_{res}	$\mathrm{MS(res)} = \frac{\mathrm{SS(res)}}{d_{\mathrm{res}}}$	
	total	$\mathrm{SS}(F)$	d_F		

that for any α and β in Ω the covariance of the corresponding response variables Y_α and Y_β only depends on the finest factor $F \in \mathcal{F}$ for which $F(\alpha) = F(\beta)$. If $\alpha = \beta$, then the finest such factor is E and the assumption implies that all response variables have the same variance σ^2. Furthermore, if $\alpha \neq \beta$ and if F is the finest plot factor in $\mathcal{F}$ with $F(\alpha) = F(\beta)$, then the covariance of Y_α and Y_β can be written as $\rho_F \sigma^2$, where ρ_F is the correlation between the two response variables. By contrast, the treatment factors are assumed to have fixed effects.

Under these assumptions the anova table for an orthogonal design $(\mathcal{F}, \mathcal{G}, T)$ can be easily constructed from the Hasse diagrams. For every plot factor in $\mathcal{F}$, that is every vertex in the Hasse diagram for plot factors, there is a stratum which is labelled by the plot factor. For the purpose of presenting only how the anova table is generated, the definition of the strata can be omitted, but more details can be found in Bailey (2008, p. 196). If $F \in \mathcal{F}$, then the corresponding degrees of freedom d_F and the sum of squares $\mathrm{SS}(F)$ for the stratum F are obtained from the relevant Hasse diagram for plot factors. Regarding the treatment factors, Theorem 10.11 in Bailey (2008, p. 198) shows that every $G \in \mathcal{G}$ is contained in a uniquely determined stratum $F \in \mathcal{F}$. More precisely, for every $G \in \mathcal{G}$ the stratum containing the source of variation for G is the one for the coarsest factor $F \in \mathcal{F}$ with $F \preceq G \circ T$. The degrees of freedom d_G and the sum of squares $\mathrm{SS}(G)$ for $G \in \mathcal{G}$ are obtained from the Hasse diagram for the treatment factors.

For every stratum the anova table then contains several rows. Table 1 shows such a portion for a stratum $F \in \mathcal{F}$, where for simplicity of presentation it is assumed that stratum F contains the treatment factors $G_1, \ldots, G_k \in \mathcal{G}$. The table has a row for each treatment factor and two additional rows for the residual in the stratum and the total. From left to right the columns give the name of the stratum, indicate sources of variation and report sums of squares, degrees of freedom, means squares and F values as appropriate.

As already mentioned, the degrees of freedom and sums of squares for F and $G_1, \ldots, G_k$ come from the Hasse diagrams. The degrees of freedom d_{res} for the residual in stratum F are obtained by subtracting the sum of the degrees of freedom of the treatment factors in the stratum from d_F. Likewise, the sum of squares SS(res) for the residual in stratum F is obtained by subtracting from SS(F) the sums of squares for the treatment factors in F. Mean squares are obtained in the usual way. In order to test the effect of a treatment factor G_j in stratum F, an F statistic is calculated as the ratio of the mean squares for G_j and the residual in F. With the additional assumption that the effects of all plot factors in $\mathcal{F}$ follow independent normal distributions it follows that under the null hypothesis of no G_j effect the corresponding F statistic has an F distribution with d_{G_j} and d_{res} numerator and denominator degrees of freedom.

Sometimes all degrees of freedom for a stratum are taken up by the treatment factors. Then, the degrees of freedom and the sum of squares for the residual in the stratum are equal to zero so that no corresponding mean square can be calculated and no tests of the treatment factors can be performed. In such cases, it still appears to be helpful to include a row for the residual if the stratum contains more than one treatment factor, because this emphasizes that no tests of treatment factors can be done. On the other hand, if the degrees of freedom for a stratum F are completely accounted for by a single treatment factor G, then F and G are confounded and the effect of G cannot be tested, but the lines for the residual in the stratum F and the total are omitted.

The full anova table for an orthogonal design $(\mathcal{F}, \mathcal{G}, T)$ is obtained by combining the portions for the individual strata $F \in \mathcal{F}$ in Table 1. By convention, subtables for coarser factors in $\mathcal{F}$ are written above the portions for finer factors. Thus, similar to the Hasse diagram for the plot factors, the full anova table has the stratum for the universal factor U on Ω at the top and the stratum for the equality factor E on Ω at the bottom. At the end of the table a line 'Total' is added, which in the sums of squares column reports the squared Euclidean length of the data vector and in the degrees of freedom column the total number of observations N. Of course, both these values can also be obtained by adding up the values in the columns above them.

Example 1 (continued) In order to illustrate the construction of the anova table we return to the experiment on apple trees where $\mathcal{F} = \{E, B, U\}$ and $\mathcal{G} = \{E, S, P, U\}$. The response data are reported by Bailey (2008, p. 145). Every factor in $\mathcal{F}$ defines a stratum and in order to be able to write down the rows in Table 1 for each of these, the treatment factors in $\mathcal{G}$ need to be allocated to E, B and U in $\mathcal{F}$, respectively.

Starting with the coarsest factor U on Ω it can be seen that, as always, this coincides with the composition of U on $\mathcal{T}$ and the design function T, which shows that the only source of variation in the U stratum is U on $\mathcal{T}$. Also,

Table 2
Full anova table for Example 1

Stratum	Source	SS	DF	MS	F
U	U	51915.9680	1	51915.9680	
B	S	1116.7461	2	558.3731	2.3321
	residual	718.2938	3	239.4313	
	total	1835.0399	5		
E	P	1835.1540	4	458.7885	6.1969
	$S \wedge P$	284.1319	8	35.5165	0.4797
	residual	888.4186	12	74.0349	
	total	3007.7046	24		
Total		56758.7125	30		

following the earlier remarks, no lines for residual or the total are shown in this stratum. The next stratum corresponds to the block factor B. Because the spray treatments are applied to entire blocks it follows that $B \preceq S \circ T$. Also, as always, for E on Ω it holds that $E \preceq S \circ T$. Figure 1 shows that B is coarser than E, and so B is the coarsest plot factor which is finer than or equivalent to $S \circ T$, which implies that S is tested in stratum B. For the pruning factor P the equality factor E on Ω is the only factor in $\mathcal{F}$ which is finer than or equivalent to $P \circ T$, and so P is tested in the final stratum for E. Similarly, it can be seen that the equality factor $E = S \wedge P$ on $\mathcal{T}$, which represents the interaction between spray treatments and pruning methods, is also tested in the stratum for E on Ω. Both the tables for the B and the E stratum contain a line for the residual. Putting together the subtables for the strata and adding the line for 'Total' then gives the full anova table in Table 2.

In order to test for effects of the treatment factors, the F values in Table 2 are compared with percentage points of appropriate F distributions. The denominator degrees of freedom of these distributions are those for the residual in the stratum in which a factor is tested. For example, the test for differences between the spray treatments is carried out in the blocks stratum and so the F distribution with 2 and 3 degrees of freedom is used, whereas both the pruning methods and the interaction of spray treatments and pruning methods are tested in the E stratum so that both tests use F distributions with 12 denominator degrees of freedom. □

The sums of squares in the full anova table, and hence the mean squares and F values, can only be computed after data have been collected. By contrast, an orthogonal design $(\mathcal{F}, \mathcal{G}, T)$ completely determines the degrees of freedom of the factors and the residuals in the different strata, even before data are available. The corresponding subtable of the full anova table which only contains the columns for the strata, sources of variation and degrees of freedom is known as the skeleton anova (Nelder, 1965b; Bailey, 2008). This table is particularly useful at the planning stage of an experiment, because it shows in which stratum each treatment factor is tested. Moreover, for each test the residual degrees of freedom in the appropriate stratum of the skeleton anova represent the denominator degrees of freedom of the relevant F distribution. By looking at these one can, for example, identify the tests with the lowest power and modify the design if the power is deemed to be too low. Thus by using the design alone the appropriate analysis can be anticipated and if necessary the design can be modified to help prevent problems which cannot be corrected once the data have been collected.

If, in addition to the design, response data are available, the *AutomaticAnova* package produces the full anova table. Otherwise, the program generates the skeleton anova from the design. It is now time to look at how the package works.

3 Automating theory: The *AutomaticAnova* algorithm

At the end of his paper, Nelder (1965b) articulated the vision of creating a computer program for analyzing simple orthogonal block structures. With contributions from, among others, Wilkinson (1970), Wilkinson and Rogers (1973) and Payne and Wilkinson (1977) this vision became a reality and continues to live on in GenStat.

This section explores the hidden algorithmic content of Bailey's theory for orthogonal designs and explains in more detail how this can be exploited for automating the anova. I start by presenting the main idea from a bird's eye perspective and then move on to addressing the more technical issues. Readers who have skipped Section 2 may consult this material as appropriate and may, in particular, wish to remind themselves about the definition of orthogonal designs.

3.1 The main algorithm

In what follows it is assumed that the experimental layout, that is the allocation of the observational units to the factors, is available in the form of a spreadsheet with a row for each observational unit and a column for each factor. Moreover, it is assumed that it has been decided which columns represent plot and treatment factors, respectively. The important consequences of these assumptions are that the following are known: (a) the set Ω of observational units, (b) the set $\tilde{\mathcal{F}}$ of plot factors on Ω and (c) the value $(G \circ T)(\omega)$ for every $G \in \tilde{\mathcal{G}}$ and every $\omega \in \Omega$, where $\tilde{\mathcal{G}}$ is the set of treatment factors and T is the design function. The set $\mathcal{T}$ of treatments is not known and, in general, cannot be deduced from the spreadsheet. This has the subtle technical implication that although $G \circ T$ is known for every $G \in \tilde{\mathcal{G}}$ as a factor on Ω, strictly speaking both T, as a function taking values in $\mathcal{T}$, and $\tilde{\mathcal{G}}$, as a set of factors defined on $\mathcal{T}$, are also not known. In what follows, I make the the mild assumption that every treatment in $\mathcal{T}$ is applied to at least one unit in Ω, so that T is a surjective mapping. Furthermore, in order to avoid clumsy terminology, in what follows the (known) set $\tilde{\mathcal{G}}_\Omega = \{G \circ T : G \in \tilde{\mathcal{G}}\}$ will also be referred to as the set of treatment factors, but from the notation it should be clear that technically this is different from $\tilde{\mathcal{G}}$.

Despite the fact that $\mathcal{T}$, T and $\tilde{\mathcal{G}}$ are not known, they do exist and for the triple $(\tilde{\mathcal{F}}, \tilde{\mathcal{G}}, T)$ one may wonder which of the properties in the definition of an orthogonal design can be checked by using only the sets $\tilde{\mathcal{F}}$ and $\tilde{\mathcal{G}}_\Omega$, which are obtained from the spreadsheet. Obviously, checking the properties under (i) and (iii) in the definition is possible. Since by assumption T is surjective, the properties (ii)(a) and (ii)(b) can also be checked. The former test amounts to verifying if the set $\tilde{\mathcal{G}}_\Omega$ contains the universal factor U on Ω. Property (ii)(b) holds if the supremum of any two elements of $\tilde{\mathcal{G}}_\Omega$ is also in the set, because, as is not difficult to prove, $(F \circ T) \vee (G \circ T) = (F \vee G) \circ T$ for any factors F and G defined on $\mathcal{T}$. The only property of an orthogonal design that cannot be checked by using $\tilde{\mathcal{F}}$ and $\tilde{\mathcal{G}}_\Omega$ from the spreadsheet is the orthogonality condition (ii)(c) on $\mathcal{T}$. However, none of the proofs in Bailey (2008) which lead to the anova described in Section 2.2 uses the property (ii)(c). Therefore, if by using $\tilde{\mathcal{F}}$ and $\tilde{\mathcal{G}}_\Omega$ it can be verified that the triple $(\tilde{\mathcal{F}}, \tilde{\mathcal{G}}, T)$ satisfies the properties (i), (ii)(a), (ii)(b) and (iii), then the same anova is obtained.

The overarching principle behind the *AutomaticAnova* algorithm presented below is as follows: If a 'user' can provide the experimental layout, that is the allocation of the observational units to the plot and treatment factors, giving rise to sets $\tilde{\mathcal{F}}$ and $\tilde{\mathcal{G}}_\Omega$ of plot and treatments factors as above, then this design information essentially determines how the analysis should be done. The correct anova is obtained by embedding the sets $\tilde{\mathcal{F}}$ and $\tilde{\mathcal{G}}_\Omega$ into the, in a sense, smallest design which satisfies all the properties of an orthogonal design except (ii)(c), provided such a design exists, and by performing the anova computations as in Section 2.2. Both the embedding and the analysis can be done automatically, but it is the embedding which makes the use of a model formula dispensable. Put differently, embedding the sets of plot and treatment factors provided by the user into a design as suggested is equivalent to inferring an appropriate model from the design. This embedding is the essence of the proposed automation of the anova and the unique feature which distinguishes the *AutomaticAnova* package from other anova software. I now present the general algorithm, which is followed by some further comments and explanations.

Assume a set $\tilde{\mathcal{F}}$ of mutually inequivalent plot factors and a set $\tilde{\mathcal{G}}_\Omega = \{G \circ T : G \in \tilde{\mathcal{G}}\}$ of mutually inequivalent treatment factors on Ω are derived from the experimental layout in a spreadsheet, where $\tilde{\mathcal{G}}$ denotes the (unknown) set of treatment factors on the (unknown) set of treatments $\mathcal{T}$. Also suppose that the (unknown) design function T is surjective.

As will be explained in more detail below, the set $\tilde{\mathcal{G}}^*_\Omega$ to be defined next is related to testing interactions. Let k be a positive integer and let $\tilde{\mathcal{G}}^*_\Omega$ be the union of $\tilde{\mathcal{G}}_\Omega$ and the set of all infima of up to k different factors in $\tilde{\mathcal{G}}_\Omega$. For example, if $\tilde{\mathcal{G}}_\Omega = \{A \circ T, B \circ T, C \circ T\}$ and $k = 3$ then $\tilde{\mathcal{G}}^*_\Omega$ contains the original factors $A \circ T$, $B \circ T$, $C \circ T$ and the infima $(A \circ T) \wedge (B \circ T)$, $(A \circ T) \wedge (C \circ T)$, $(B \circ T) \wedge (C \circ T)$ and $(A \circ T) \wedge (B \circ T) \wedge (C \circ T)$. If some of these factors are equivalent, then for each subset of equivalent factors $\tilde{\mathcal{G}}^*_\Omega$ only contains one of these factors. For $k = 1$ it follows that $\tilde{\mathcal{G}}^*_\Omega = \tilde{\mathcal{G}}_\Omega$. It is not difficult to prove that $(F \circ T) \wedge (G \circ T) = (F \wedge G) \circ T$ for any factors F and G defined on $\mathcal{T}$. This implies that forming $\tilde{\mathcal{G}}^*_\Omega$ is equivalent to adding to $\tilde{\mathcal{G}}$ all infima of up to k factors in $\tilde{\mathcal{G}}$ and forming the composition $H \circ T$ for every factor H in the augmented set. The *AutomaticAnova* algorithm can then be described as follows.

Algorithm 1 *Suppose a set of mutually inequivalent plot factors $\tilde{\mathcal{F}}$, a set of mutually inequivalent treatment factors $\tilde{\mathcal{G}}_\Omega$ and a positive integer k are given.*

- *(A1) Find the the smallest set $\mathcal{F}$ of factors on Ω with $\tilde{\mathcal{F}} \subseteq \mathcal{F}$ which includes the factors U and E on Ω, and which is such that for every $F, G \in \mathcal{F}$ the infimum $F \wedge G$ and the supremum $F \vee G$ are also in $\mathcal{F}$.*
- *(A2) Form the set $\tilde{\mathcal{G}}^*_\Omega$ and find the smallest set $\mathcal{G}_\Omega$ of factors on Ω with $\tilde{\mathcal{G}}^*_\Omega \subseteq \mathcal{G}_\Omega$ which includes U on Ω and which is such that for every $F, G \in \mathcal{G}_\Omega$ the supremum $F \vee G$ is also in $\mathcal{G}_\Omega$.*
- *(A3) Check if $\mathcal{F}$ fulfills the conditions (i)(a) and (i)(f) for an orthogonal design. If any of these is violated, terminate with an error message.*
- *(A4) For $\mathcal{G}_\Omega$ check the orthogonality requirement (iii)(a) in the definition of an orthogonal design and (iii)(b) for $\mathcal{F}$ and $\mathcal{G}_\Omega$. If any of these is violated, terminate with an error message.*
- *(A5) For every $F \in \mathcal{F}$ and every $G \in \mathcal{G}_\Omega$ whose supremum $F \vee G$ is not in $\mathcal{G}_\Omega$ add $F \vee G$ to $\mathcal{G}_\Omega$.*
- *(A6) Carry out the anova with plot factors $\mathcal{F}$ from (A1) and treatment factors $\mathcal{G}_\Omega$ from (A5) as described in Section 2.2.*

The following additional comments about the algorithm may be helpful. In practice, the set $\mathcal{F}$ in (A1) is obtained by adding to $\tilde{\mathcal{F}}$ the infimum $F \wedge G$ and the supremum $F \vee G$ of every $F, G \in \tilde{\mathcal{F}}$ and iterating the process until

$\tilde{\mathcal{F}}$ becomes stable. Subsequently, unless they are already in $\tilde{\mathcal{F}}$, the factors U and E on Ω are also included. If in the process of adding infima and suprema some factors turn out to be equivalent, then for every such set of equivalent factors only a single factor is included in the final set $\mathcal{F}$. If the tests in (A3) are successful, then $\mathcal{F}$ possesses the properties (i) of an orthogonal design.

The way in which the set $\mathcal{G}_\Omega$ in (A2) is obtained is more complicated and requires some further explanation. By assumption T is surjective. This implies that adding U on Ω to the set $\tilde{\mathcal{G}}_\Omega = \{G \circ T : G \in \tilde{\mathcal{G}}\}$ is equivalent to adding U on $\mathcal{T}$ to $\tilde{\mathcal{G}}$. Likewise, since $(F \circ T) \vee (G \circ T) = (F \vee G) \circ T$ for any factors F and G defined on $\mathcal{T}$, adding the supremum $(F \circ T) \vee (G \circ T)$ of any two factors in $\tilde{\mathcal{G}}_\Omega$ to $\tilde{\mathcal{G}}_\Omega$ is equivalent to adding the supremum $F \vee G$ of $F, G \in \tilde{\mathcal{G}}$ to $\tilde{\mathcal{G}}$. Thus adding $(F \circ T) \vee (G \circ T)$ for every $F \circ T, G \circ T \in \tilde{\mathcal{G}}_\Omega$ to $\tilde{\mathcal{G}}_\Omega$, iterating the process until $\tilde{\mathcal{G}}_\Omega$ does not change any more, and finally, unless the factor is already included, adding U on Ω to $\tilde{\mathcal{G}}_\Omega$ ensures that at the end of the process the underlying set $\tilde{\mathcal{G}}$ of treatment factors on $\mathcal{T}$ satisfies the properties (ii)(a) and (ii)(b) of an orthogonal design. In a similar way, in step (A2) the set $\mathcal{G}_\Omega$ is obtained by adding to $\tilde{\mathcal{G}}^*_\Omega$ all suprema $(F \circ T) \vee (G \circ T)$ of factors in $\tilde{\mathcal{G}}^*_\Omega$, iterating the process and, if not already included, adding U on Ω. This procedure ensures that $\tilde{\mathcal{G}}^*_\Omega \subseteq \mathcal{G}_\Omega$ and that every factor in $G \in \mathcal{G}_\Omega$ can be represented as $G = \bar{G} \circ T$ where $\bar{G}$ is from a set of treatment factors $\mathcal{G}$ on $\mathcal{T}$ which satisfies the requirements (ii)(a) and (ii)(b) for an orthogonal design. It should be noted that since every factor $G \in \mathcal{G}_\Omega$ can be written as $G = \bar{G} \circ T$ the tests of properties (iii)(a) and (iii)(b) in (A4) are well-defined.

The reason for generating $\mathcal{G}_\Omega$ from $\tilde{\mathcal{G}}^*_\Omega$ and not from the original $\tilde{\mathcal{G}}_\Omega$ is that after adding suprema the augmented set $\tilde{\mathcal{G}}_\Omega$ will not necessarily contain the infima $(F \circ T) \wedge (G \circ T)$ of factors $F \circ T, G \circ T$ in the original $\tilde{\mathcal{G}}_\Omega$ from the spreadsheet which are such that neither $(F \circ T) \prec (G \circ T)$ nor $(G \circ T) \prec (F \circ T)$. In order to be able to test the interaction between any two such factors it is however necessary to add the infimum for every $F \circ T$ and $G \circ T$ in the original set $\tilde{\mathcal{G}}_\Omega$ where F is not finer than G and vice versa to $\tilde{\mathcal{G}}_\Omega$. This is done by forming the set $\tilde{\mathcal{G}}^*_\Omega$ with $k = 2$. Similarly, if interactions involving up to $k > 2$ factors in $\tilde{\mathcal{G}}_\Omega$ are to be tested, the set $\tilde{\mathcal{G}}^*_\Omega$ for the desired value of k needs to be constructed from $\tilde{\mathcal{G}}_\Omega$. Thus by making k part of the input, the algorithm offers the flexibility to include, up to a certain degree, all interactions of the treatment factors in the input set $\tilde{\mathcal{G}}_\Omega$ in the anova.

One not immediately obvious consequence of the choice of k in the input to the algorithm is worthwhile to note. When forming $\tilde{\mathcal{G}}^*_\Omega$, all infima involving *up to* k of the factors in $\tilde{\mathcal{G}}_\Omega$ are included, thereby enabling anova tests of interactions involving up to k factors. However, if, for example, $k = 3$ and the infimum $(F \circ T) \wedge (G \circ T) \wedge (H \circ T)$ of three factors in $\tilde{\mathcal{G}}_\Omega$ is equivalent to either one of the original factors in $\tilde{\mathcal{G}}_\Omega$ or to the infimum of two factors in $\tilde{\mathcal{G}}_\Omega$, then $(F \circ T) \wedge (G \circ T) \wedge (H \circ T)$ will not be added to $\tilde{\mathcal{G}}^*_\Omega$ and the corresponding three-factor interaction cannot be tested. In other words, for every value of k the algorithm attempts to include as many as possible tests of interactions involving at most k treatment factors, but only performs the tests which the input design as represented by $\tilde{\mathcal{F}}$ and $\tilde{\mathcal{G}}_\Omega$ allows to be done.

After completing step (A5) the sets $\mathcal{F}$ and $\mathcal{G}_\Omega$ satisfy the property (iii)(c) of an orthogonal design. This follows since, as explained above, every factor $G \in \mathcal{G}_\Omega$ can be written as $\bar{G} \circ T$, where $\bar{G}$ is a factor on $\mathcal{T}$, and adding the supremum $F \vee G = F \vee (\bar{G} \circ T)$ for $F \in \mathcal{F}$ to $\mathcal{G}_\Omega$ is, by using that T is surjective, equivalent to adding a factor H on $\mathcal{T}$ with $H \circ T = F \vee (\bar{G} \circ T)$ to the set $\mathcal{G}$ of treatment factors on $\mathcal{T}$. The discussion in Bailey (2008, p. 197) implies that step (A5) does not have to be iterated. That is, if all necessary suprema $F \vee G$ of factors $F \in \mathcal{F}$ and $G \in \mathcal{G}_\Omega$ have been added to $\mathcal{G}_\Omega$, then there is no need to repeat (A5) with $\mathcal{F}$ and the augmented set $\mathcal{G}_\Omega$. Likewise, it is not necessary to repeat the test of property (ii)(b).

Moreover, it follows from Bailey (2008, p. 198) that if in (A4) the test of property (iii)(a) confirms the orthogonality of the factors in $\mathcal{G}_\Omega$ and the test of (iii)(b) the orthogonality of the factors in $\mathcal{F}$ and $\mathcal{G}_\Omega$, then at the end of (A5) the orthogonality conditions (iii)(a) for the augmented set $\mathcal{G}_\Omega$ and (iii)(b) for $\mathcal{F}$ and the augmented set $\mathcal{G}_\Omega$ are also satisfied. In practice this means that the conditions (i)(f) for $\mathcal{F}$ from (A1) and (iii)(a) for $\mathcal{G}_\Omega$ from (A2) are checked separately and that subsequently (iii)(b) is tested for $\mathcal{F}$ and $\mathcal{G}_\Omega$. If all these tests confirm the orthogonality, then the suprema necessary for (iii)(c) are added to $\mathcal{G}_\Omega$ in step (A5) without having to repeat the tests of (iii)(a) and (iii)(b) for the augmented set $\mathcal{G}_\Omega$.

If the tests in steps (A3) and (A4) of the algorithm are successful, then the design represented by $\mathcal{F}$ from (A1) and the augmented set $\mathcal{G}_\Omega$ from (A5) satisfies all the requirements of an orthogonal design apart from (ii)(c). Therefore, in the final step (A6) the anova can be performed as described in Section 2.2.

3.2 Algorithmic details

Algorithm 1 represents a top-level blueprint which describes the main tasks to be carried out by the *AutomaticAnova* program. However, there are several additional issues which need to be considered at a more detailed level. Below, these are addressed in an order which roughly corresponds to the sequence in which the different issues need to be dealt with in a piece of computer code.

Although the distinction between plot and treatment factors needs to be maintained in order to be able to derive the correct anova, in practice all factors in this section are defined on Ω. In what follows, every factor F with n_F levels on Ω will be identified with the set of its F-classes. That is, the letter F refers to both the factor name and the set $\{F[[i]] : i = 1, \ldots, n_F\}$.

3.2.1 Finding infima and suprema

Let F and G be two factors defined on the same set with n_F and n_G levels, respectively. As indicated in Section 2, finding the infimum of F and G is easy, and the $F \wedge G$-classes are the non-empty intersections of the F-classes and G-classes (Bailey, 2008, p. 172), that is $F \wedge G = \{F[[i]] \cap F[[j]] : F[[i]] \cap F[[j]] \neq \emptyset, i = 1, \ldots, n_F, j = 1, \ldots, n_G\}$.

Finding the supremum $F \vee G$ is more involved. Bailey (2008, p. 173) describes one possibility, which however does not translate easily into suitable computer code. As was noted in Section 2 every $F \vee G$-class can be represented as the union of some F-classes and also as the union of some G-classes. The class of $F \vee G$ containing any given F-class $F[[i]]$, say, can then be obtained by joining all other F-classes $F[[j]]$ with $F[[i]]$, for which there exists a G-class intersecting both $F[[j]]$ and $F[[i]]$, and iterating this process as appropriate. In other words, if $F[[j]]$ is joined with $F[[i]]$ the logic has to be repeated for $F[[j]]$. These considerations lead to the algorithm below, which describes the procedure by using standard set operations and pseudo-code for assignments and flow control. This is to be used on copies of F and G so that the original factors are still available after $F \vee G$ has been computed.

Algorithm 2 *Suppose that $F = \{F[[i]] : i = 1, \ldots, n_F\}$ and $G = \{G[[j]] : j = 1, \ldots, n_G\}$ are factors on the same set and that the F-classes in F are arranged in an arbitrary but fixed order. Denote by $F_{(i)}$ the ith element of F. This notation is introduced, because the size of F shrinks during the execution of the algorithm. Then find $F \vee G$ as follows:*

```
$F \vee G := \emptyset$;
While $|F| > 0$ do
    $(F \vee G)_{new} := F_{(1)}$;
    $F := F \setminus \{F_{(1)}\}$;
    $i := |F|$;
    While $i > 0$ do
        If exists $G[[j]] \in G : (F \vee G)_{new} \cap G[[j]] \neq \emptyset$ and $F_{(i)} \cap G[[j]] \neq \emptyset$
            Then $(F \vee G)_{new} := (F \vee G)_{new} \cup F_{(i)}$;
                 $F := F \setminus \{F_{(i)}\}$;
                 $i := |F|$;
            Else $i := i - 1$;
        End If;
    End While;
    $F \vee G := F \vee G \cup \{(F \vee G)_{new}\}$;
End While;
```

Return $F \vee G$;

It is worth noting that both the infimum and the supremum can be computed without having to refer to the partial order $\preceq$ used in their definitions.

3.2.2 The closure of a set of factors

In order to embed the input set $\tilde{\mathcal{F}}$ of plot factors into a set $\mathcal{F}$ of factors on Ω which has the properties (i)(b)–(i)(e) of an orthogonal design, unless they are already in $\tilde{\mathcal{F}}$ the universal factor U and the equality factor E have to be added. Moreover, for every $F, G \in \tilde{\mathcal{F}}$ the infimum $F \wedge G$ and the supremum $F \vee G$ have to be included. There is a tricky issue here, because if at some point, for example, $F \vee G$ is added, then it needs to be made sure that properties (i)(d) and (i)(e) hold for the augmented set. This implies that the process of including infima and suprema has to be iterated until no new infima or suprema are added. For convenience, I refer to the resulting set $\mathcal{F}$ as the closure of $\tilde{\mathcal{F}}$.

In practice, the closure $\mathcal{F}$ with the required properties (i)(b)–(i)(e) can be obtained by using the following algorithm. Again, this is meant to be carried out on a copy of $\tilde{\mathcal{F}}$, so that the original set $\tilde{\mathcal{F}}$ remains unchanged.

Algorithm 3 *Set $\mathcal{F}$ equal to $\tilde{\mathcal{F}}$. Then, until $\mathcal{F}$ stops changing repeat the following steps:*

(1) For every pair $F, G \in \tilde{\mathcal{F}}$, add $F \wedge G$ and $F \vee G$ to $\mathcal{F}$, provided these factors are not already elements of $\mathcal{F}$.
(2) Set $\tilde{\mathcal{F}}$ equal to $\mathcal{F}$ obtained in (1).

Finally, if not already included, add U and E to $\mathcal{F}$.

In step (2), replacing $\tilde{\mathcal{F}}$ with $\mathcal{F}$ from (1) ensures, that the next time (1) is executed, it is checked if further infima or suprema, which involve some of the new factors from the most recent execution of (1), need to be added to $\mathcal{F}$. Adding U and E at the end of the procedure is computationally more efficient, but both factors could also be added to $\mathcal{F}$ before starting the iteration.

As explained in Section 3.1, in order to be able to test interactions involving up to k treatment factors in $\tilde{\mathcal{G}}_\Omega$ from the spreadsheet, where k is part of the input, the step (A2) of the *AutomaticAnova* algorithm which involves the set $\tilde{\mathcal{G}}^*_\Omega$ is performed. In order to create $\tilde{\mathcal{G}}^*_\Omega$ from the input set $\tilde{\mathcal{G}}_\Omega$ for the given k, initially $\tilde{\mathcal{G}}^*_\Omega$ is set equal to $\tilde{\mathcal{G}}_\Omega$. Subsequently, for every $(F \circ T) \in \tilde{\mathcal{G}}^*_\Omega$ and every $(G \circ T) \in \tilde{\mathcal{G}}_\Omega$, unless it is already included, the infimum $(F \circ T) \wedge (G \circ T)$ is added to $\tilde{\mathcal{G}}^*_\Omega$. In total, this step is carried out $k - 1$ times which ensures, that every infimum in the final set $\tilde{\mathcal{G}}^*_\Omega$ involves at most k of the original factors in

$\tilde{\mathcal{G}}_\Omega$.

Embedding $\tilde{\mathcal{G}}^*_\Omega$ into a set $\mathcal{G}_\Omega$ in step (A2) requires an iterative procedure similar to Algorithm 3. However, in the step corresponding to (2) above only suprema are added. Also only U is included at the end. Again, it is convenient to call the resulting set $\mathcal{G}_\Omega$ the closure of $\tilde{\mathcal{G}}^*_\Omega$.

The input sets $\tilde{\mathcal{F}}$ and $\tilde{\mathcal{G}}_\Omega$ to Algorithm 1 consist of mutually inequivalent factors. The procedures in this section ensure that this property carries over to the closures $\mathcal{F}$ and $\mathcal{G}_\Omega$. It is however possible that $F \equiv (G \circ T)$ for some $F \in \mathcal{F}$ and $(G \circ T) \in \mathcal{G}_\Omega$.

3.2.3 Extracting the $\prec$ and $\preceq$ relations

Although not necessarily required for computing infima and suprema, the finer relation $\prec$ and the finer or equivalent relation $\preceq$ are needed for generating Hasse diagrams and for the anova calculations. Also, having a concise representation of the relations is helpful when checking if factors are orthogonal.

Let $\mathcal{H}$ be some set of m factors defined on the same set, for example, one of the sets $\mathcal{F}$ or $\mathcal{G}_\Omega$ in Algorithm 1. Also assume that the factors in $\mathcal{H}$ are arranged in an arbitrary but fixed order. The $\preceq$ relation for the factors in $\mathcal{H}$ can be represented by an $m \times m$ matrix with elements equal to zero or one. By interpreting the row and column numbers as the positions of the factors in $\mathcal{H}$, every element of the matrix can be associated with an ordered pair of factors in $\mathcal{H}$. If $F \preceq G$, then the corresponding element of the matrix will be equal to one; otherwise it is zero. Because every factor can be regarded as a collection of its classes, it can be decided if $F \preceq G$ by checking for every F-class if it is a subset of some G-class. If $F \preceq G$ and at least one of the F-classes is a proper subset of one of the G-classes, then $F \prec G$, which shows how a similar matrix representing the finer relation can be obtained.

If the factors in $\mathcal{H}$ are mutually inequivalent, then the $\preceq$ relation coincides with the $\prec$ relation, except that for every $F \in \mathcal{H}$ it holds that $F \preceq F$ but not $F \prec F$. Thus for mutually inequivalent factors a matrix representing the $\prec$ relation can be obtained from the matrix for $\preceq$ by replacing all diagonal elements with zero.

If in addition to $\mathcal{H}$ there is another ordered set $\mathcal{I}$ of n factors which are defined on the same set as those in $\mathcal{H}$, then by using an $m \times n$ matrix where the rows correspond to factors in $\mathcal{H}$ and the columns to factors in $\mathcal{I}$ the relation $F \preceq G$ for $F \in \mathcal{H}$ and $G \in \mathcal{I}$ can be represented similarly. In the *AutomaticAnova* algorithm this relation is considered for $\mathcal{H} = \mathcal{F}$ and $\mathcal{I} = \mathcal{G}_\Omega$.

3.2.4 Hasse diagrams

For a set $\mathcal{H}$ of m mutually inequivalent factors on the same set which are arranged in an arbitrary but fixed order, obtaining the directed graph underlying the Hasse diagram from the matrix for the finer relation $\prec$ in Section 3.2.3 is straightforward. This can be done by computing the edge set or, equivalently, the adjacency matrix of the graph.

For $F, G \in \mathcal{H}$ there is an edge from F to G if and only if (a) $F \prec G$ and (b) there exists no other $H \in \mathcal{H}$ with $F \prec H$ and $H \prec G$. Thus if F is the ith and G is the jth factor in $\mathcal{F}$, where $i \neq j$, then there is an edge between the two factors if the (i, j)th element of the $\prec$ matrix is equal to one and if for every h different from both i and j the product of the (i, h)th and (h, j)th elements of the matrix is zero.

The adjacency matrix is an $m \times m$ matrix whose row and column numbers can be interpreted as the positions of the factors in $\mathcal{F}$. The (i, j)th element of the matrix is equal to one if there is an edge between the factors at positions i and j in $\mathcal{F}$, and zero otherwise. In step (A6) of Algorithm 1 this procedure is applied separately to $\mathcal{H} = \mathcal{F}$ and $\mathcal{H} = \mathcal{G}_\Omega$. For drawing the Hasse diagram I assume that a routine is available which can generate the plot from the set of edges or the adjacency matrix.

3.2.5 Checking orthogonality and balance

In what follows suppose that $\mathcal{F}$ is the closure of $\tilde{\mathcal{F}}$ and $\mathcal{G}_\Omega$ the closure of $\tilde{\mathcal{G}}_\Omega^*$ from Section 3.2.2. In order to check the orthogonality requirements (i)(f), (iii)(a) and (iii)(b) for $\mathcal{F}$ and $\mathcal{G}_\Omega$ it is possible to implement a literal test of the defining condition for orthogonality in Section 2.1. The required amount of computation can be reduced if $F \preceq G$, because then F and G are orthogonal (Bailey, 2008, p. 179) and the more elaborate test can be avoided.

Checking the condition (i)(a) in step (A3) for the plot factors in $\mathcal{F}$ is straightforward and amounts to verifying that for every $F \in \mathcal{F}$ the F-classes have the same size.

3.2.6 Keeping track of factor names

The factors in the input sets $\tilde{\mathcal{F}}$ and $\tilde{\mathcal{G}}_\Omega$ to Algorithm 1 are assumed to have names which are provided by the user, for example as the column headings in a spreadsheet. So far for convenience the factors have been called F, G and so forth, but in applications they will usually carry more meaningful names. During the execution of the *AutomaticAnova* algorithm new factors are created from the original ones by means of the $\wedge$ and $\vee$ operators. Both are binary operators, but as mentioned in Section 2.1 they possess the property of being associative which implies that factors such as $F \vee G \vee H \vee I$ are well-defined. Keeping track of the correct factor names requires some careful bookkeeping. For example, if the supremum of $A \wedge B$ and $F \vee G$ is formed where $B \prec F$, then $(A \wedge B) \vee (F \vee G) = F \vee G$ since $A \wedge B \preceq B \prec F \preceq F \vee G$ and so the simpler name $F \vee G$ for the supremum should be used.

3.2.7 Anova computations

If for the given inputs $\tilde{\mathcal{F}}$, $\tilde{\mathcal{G}}_\Omega$ and k Algorithm 1 has been successful in embedding $\tilde{\mathcal{F}}$ and $\tilde{\mathcal{G}}_\Omega$ into sets $\mathcal{F}$ and $\mathcal{G}_\Omega$, then in step (A6) the anova table is constructed as described in Section 2.2. If response data are available, then the full anova table is produced which for convenience also reports p-values for the different F tests. Otherwise, the algorithm returns the skeleton anova table.

Calculations of degrees of freedom and sums of squares are done separately for the plot factors in $\mathcal{F}$ and the treatment factors in $\mathcal{G}_\Omega$. The crude sum of squares CSS(F) for a factor F on Ω with n_F levels is calculated as (Bailey, 2008, p. 178)

$$\mathrm{CSS}(F) = \sum_{i=1}^{n_F} \frac{(\text{total of the responses for all units in } F[[i]])^2}{\text{size of } F[[i]]},$$

where $F[[i]]$ is the F-class for level i. As explained in Section 2.2, for the residual in each stratum the degrees of freedom and sums of squares are obtained by subtraction. In order to allocate the treatment factors to the strata the $m \times n$ matrix representing the $\preceq$ relation for $\mathcal{F}$ and $\mathcal{G}_\Omega$ at the end of Section 3.2.3 is used.

4 Implementation

Algorithm 1 was implemented in MATHEMATICA 8 as the *AutomaticAnova* package. This package provides a single function for reading in the data and carrying out the anova computations. In order to enhance usability, the program was supplemented with a graphical user interface (GUI) in the *AutomaticAnovaGUI* package, which loads the *AutomaticAnova* package and provides a command for launching the GUI. Both packages were extensively tested on Microsoft Vista and Windows platforms. Their usage is explained in the next section.

Input data are read from a Microsoft Excel file containing the experimental layout in a spreadsheet with rows for the observational units and named columns for the factors. Response data are not necessarily required. From the spreadsheet input sets to Algorithm 1 of plot and treatment factors $\tilde{\mathcal{F}}$ and $\tilde{\mathcal{G}}_\Omega$ are created as specified by the user. The data structure for a factor F is a list of lists, or set of sets, which literally corresponds to the representation $\{F[[i]] : i = 1, \ldots, n_F\}$.

The implementation heavily uses set operations, such as the union and intersection, for computing infima, suprema and closures. Design matrices and matrix inversions are not required. The drawing of Hasse diagrams is accomplished by means of a built-in Mathematica function which can generate the diagram from the adjacency matrix. At each vertex the name F of a factor is displayed together with the pair (n_F, d_F) indicating the number of levels and degrees of freedom. In the Hasse diagram and the anova table the infimum $F \wedge G$ of two factors F and G is shown as 'I(F, G)' and the supremum $F \vee G$ as 'S(F, G)'. This notation extends naturally to more than two factors.

5 Package usage

The governing principle behind the design of the *AutomaticAnova* package and its GUI was the wish to create a user-friendly piece of software which would be intuitive to use and which, as far as possible, would protect the non-expert from going wrong. In order to minimize the risk of incorrect input specifications, the *AutomaticAnovaGUI* package uses dynamic enabling of dialogs and options to guide the user through the few steps which are necessary for analyzing a set of data. Since in many fields of application data are stored in Microsoft Excel files, this input format is also used by the *AutomaticAnova* and the *AutomaticAnovaGUI* packages.

Essentially, the user then has to select the plot and treatment factors and, if available, a response variable from an Excel spreadsheet. Furthermore, the maximum number of treatment factors which can be involved in tests of interaction effects has to be specified. As was mentioned before, the program does not require a model formula. The resulting anova table together with the Hasse diagrams can be saved as a nicely formatted PDF file.

The remainder of this section provides a brief user guide. The focus is on the GUI provided by the *AutomaticAnovaGUI* package, which is recommended for most users. Despite being used for the implementation, it is worth pointing out that no knowledge of Mathematica is needed.

5.1 Software requirements and package files

In order to get started the user needs to have access to Mathematica 8. In what follows, I assume that this has been installed under Microsoft Vista or Windows. The *AutomaticAnova* and *AutomaticAnovaGUI* packages are provided in the two files `AutomaticAnova.m` and `AutomaticAnovaGUI.m`, which can be downloaded from the author's web page. It is also planned to make the packages available for download as supplementary files from the journal's web page. These files need to copied to the same directory. For simplicity suppose that this directory is called `C:\MyFiles`, but note that the user is completely free to choose the directory.

5.2 Input file

The packages work on input data provided as a Microsoft Excel spreadsheet in a file with extension `.xls` or `.xlsx`. Many other programs can export data in that format and such files may also be used. The spreadsheet is expected to have a row for each observational unit and named columns for the factors and for the response variable, if present.

The cells in the column for a factor need to show the levels of the factor which are assigned to the observational units in the rows. That is, for a factor F each row contains the value $F(\omega)$ for the unit ω represented by the row. In the current implementation, the factor levels must be integers, so no text labels can be used. Usually the levels of a factor F with n_F levels are coded by the integers $1, \ldots, n_F$ but sometimes it is more convenient to use other values. For example, if a factorial design is used in which the levels of F are represented by the numbers $0, \ldots, n_F - 1$ then these values can be used directly. Also, the levels do not necessarily have to be consecutive numbers, which is useful when the levels represent selected values of a quantitative variable. Since internally every factor is identified with the set of its classes, it is however immaterial which integers are used for indicating the different factor levels. In particular, the factor levels carry no numerical or quantitative meaning and just serve to distinguish the levels of a factor from each other.

In addition there is one convention which needs to be observed and which is crucial for the program to be able to identify the correct model. This concerns the coding of the levels of nested factors. To illustrate how the coding has to be done, consider an experiment where 3 educational programmes (P) are to be compared in 7 different school districts (D) with 20 schools (S) per district and using 5 classes (C) from each school, where the letters in brackets represent abbreviated names of the corresponding factors in the spreadsheet.

Here P is a treatment factor and the other factors are plot factors. In total there are then $N = 700$ observational units and hence rows in the spreadsheet (in addition to the one for the column names).

The levels of P are coded as $1, 2, 3$. Likewise, because D is the coarsest plot factor its levels are coded by the integers $1, \ldots, 7$. In the spreadsheet, the 100 units representing the classes within the same district have the same level of D. The next finer plot factor is S. The 20 schools in each district are different from the schools in every other district, so in total there are 140 schools. It is essential to note that therefore S is treated as a factor with $n_S = 140$ levels, which are coded as $1, \ldots, 140$. Thus, for example, in the seven districts the levels of S representing the first school are 1, 21, 41, 61, 81, 101 and 121, respectively. If instead S was regarded as a factor with $n_S = 20$ levels, then the program would assume that the school represented by level 1 is the same in all school districts. Similarly, because the classes within a school are different from the classes in every other school the factor C has $n_C = 700$ levels which are represented by the integers $1, \ldots, 700$. Although in this example the coding has been explained for plot factors it is applied in the same way, if there are nested treatment factors.

In general the equality factor E and the universal factor U do not have to be part of the spreadsheet and are added internally as appropriate by the program. Since in the above example the factor C for classes coincides with E the column for C can be omitted from the spreadsheet.

The above coding of the nested factor for schools serves the same purpose as the nesting and dot operators in GenStat or some formal interaction terms in the SAS model specification, where however both programs would treat schools as a factor S with $n_S = 20$ levels only. To the non-statistician these mechanisms for telling the software that schools in one district are different from schools in other districts usually remain nebulous and incomprehensible. By contrast, it appears that the coding used here can be explained much more easily, because it corresponds to the actual conduct of an experiment. After looking at some examples, I believe, the non-expert user will understand that levels corresponding to smaller objects, such as schools, which are contained in larger objects, such as districts, have to be represented by different numbers.

5.3 Loading the packages

After starting MATHEMATICA, in order to be able to use the *AutomaticAnova* and *AutomaticAnovaGUI* packages, first one needs to open a new notebook file by using the program's File menu. Secondly, a command needs to be executed which tells MATHEMATICA where to find the `AutomaticAnova.m` and `AutomaticAnovaGUI.m` files. Assuming that these have been copied to the directory `C:\MyFiles` as suggested in Section 5.1, in the notebook the user needs to type and execute the command `SetDirectory["C:\\MyFiles"]`. Note that the directory name has to be enclosed in double quotes and that a double backslash is used to represent the single backslash in the directory name. The quickest way to execute the command is by selecting it with the mouse and simultaneously pressing the `Shift` and `Enter` keys on the keyboard.

Finally, one of the packages has to be loaded. If the GUI is to be used, then the command `Get["AutomaticAnovaGUI`"]` needs to be typed and executed in exactly the same way. The grave accent (ASCII character 96) tells Mathematica that a package file is to be loaded. Note that this command should be executed only once per Mathematica session, since it defines several names which are protected when a second attempt to load the package is made. Similarly, the command `Get["AutomaticAnova`"]` loads the command line version of the package. When loading the GUI there is no need to also execute the `Get["AutomaticAnova`"]` command, because this is handled internally.

The above steps have to repeated at the beginning of every Mathematica session. Alternatively, the notebook containing the two commands for setting the directory and loading the appropriate package can be saved as a file with extension `.nb`, which can be opened and evaluated at the start of a new session.

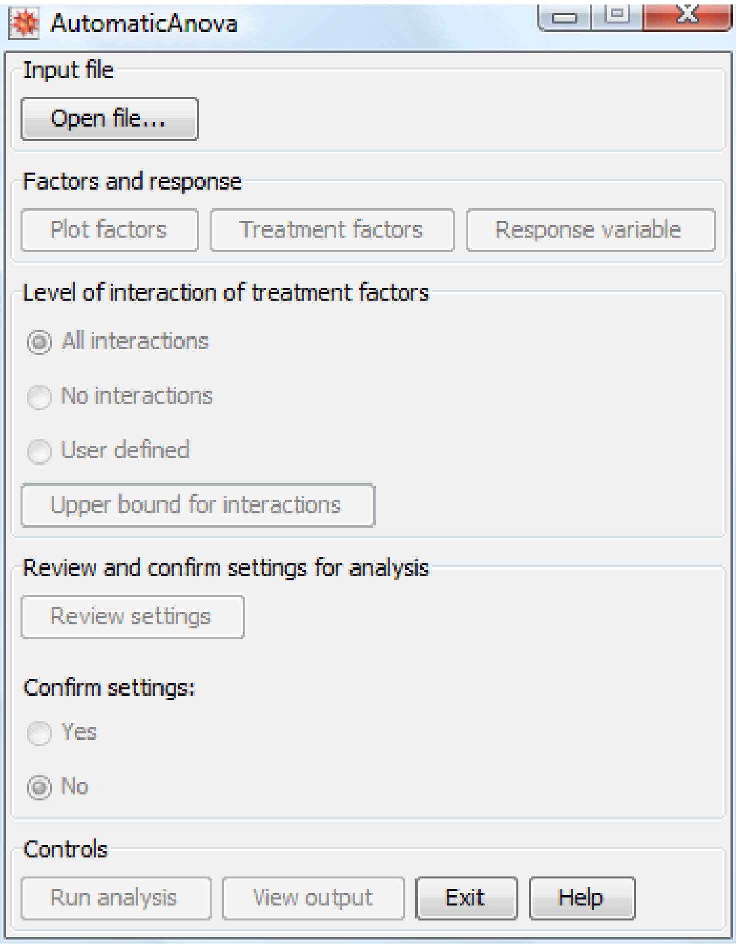
5.4 GUI workflow

After loading the *AutomaticAnovaGUI* package the GUI shown in Figure 2 is launched by typing and executing the command `AutomaticAnovaGUI[]` in the Mathematica notebook which is already open. The GUI has five sections, briefly described below, which become active at the appropriate points in the analysis process.

Input file The Microsoft Excel input file described in Section 5.2 is opened by using a standard dialog.

Factors and response Plot and treatment factors in the input file can be selected by clicking on the respective buttons. In addition, a single response variable can be specified. Each of those specifications is optional. The program produces the full anova table if all three types of variable are specified and the skeleton anova if the response variable is missing. If treatment but no plot factors are selected, then the analysis for a completely randomized design with the chosen treatment factors is performed. Conversely, if plot but no treatment factors are specified, then the anova table for a random-effects-only model is generated.

Level of interaction of treatment factors This section becomes active only



● All interactions

○ No interactions

○ User defined

Confirm settings:

○ Yes

● No

Fig. 2. Graphical user interface provided by *AutomaticAnovaGUI* package

if treatment factors were selected. Otherwise, it has no effect. By default the program tries to include all possible interactions of the treatment factors in the analysis. Alternatively, a main-effects-only analysis can be requested. Another alternative is to specify the maximum number of treatment factors for which tests of interactions are to be performed by selecting the "User defined" option and clicking on the "Upper bound for interactions" button. This upper bound specifies the input value k for Algorithm 1. Users who are not sure what the appropriate level of interaction is should, at least for the purpose of an initial analysis, make no changes to this section.

Review and confirm settings for analysis Before being able to run the anova the user must review the input specifications in a separate window by clicking on the corresponding button. If everything is correct, then the option for confirming the settings needs to be selected. Otherwise, the user can go back to the previous sections and make changes.

Controls It is only after the settings have been confirmed that the "Run analysis" button becomes active. The output is displayed in a separate window from where it can be saved as a report in a PDF file.

5.5 Command line mode

More experienced users of MATHEMATICA may sometimes wish to access the objects containing the Hasse diagrams and the anova table directly. This is possible by loading the `AutomaticAnova.m` file and executing the function `AutomaticAnova` provided therein in a MATHEMATICA notebook. A detailed description of how this function is used can be obtained by executing the command `?AutomaticAnova`. Since for the majority of users the GUI is recommended, I do not present further details here.

6 Examples

This section illustrates the use of the GUI in Section 5.4 with several examples. Although the *AutomaticAnovaGUI* package is used throughout, in what follows I use the shorter name *AutomaticAnova* as a shorthand for this. The examples range from simple to complex and illustrate the wide range of designs that can be analyzed. Limitations of the approach will be discussed at the end of the paper.

6.1 Design with between-subjects and within-subjects factors

Experiments where individuals (people or animals), usually referred to as subjects, are sequentially exposed to several treatments are common in many application areas and the corresponding factors are known as within-subjects factors. In addition, there are often between-subject factors which do only vary across but not within the individuals. The responses of an individual can be regarded as a block of observations and in the analysis this is reflected by introducing a plot factor for subjects.

Example 2 A hypothetical study presented by Gamst et al. (2008, p. 391) involves twenty college students and has two between- and one within-subjects factor. One between-subject factor is 'Age' which distinguishes between two age groups and the other is 'Gender'. Half of the participants are from each age group and in both groups equal numbers of male and female students are used. The within-subject factor 'Attraction' has two levels corresponding to descriptions of a person by whom a respondent's romantic partner is supposedly attracted. Every participant reads both descriptions and a score reflecting strength of jealousy is measured immediately after reading each version (for the data see Gamst et al., 2008, p. 392). In this simple example, the authors assume that there are no effects of the order in which the descriptions are

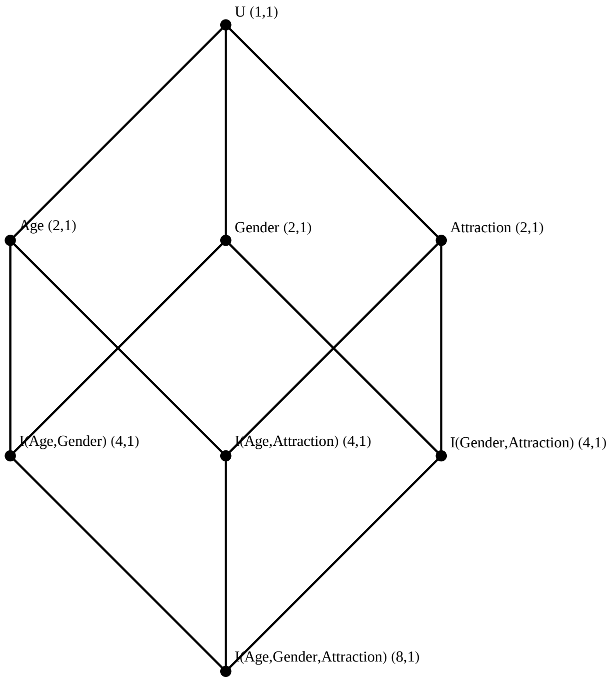
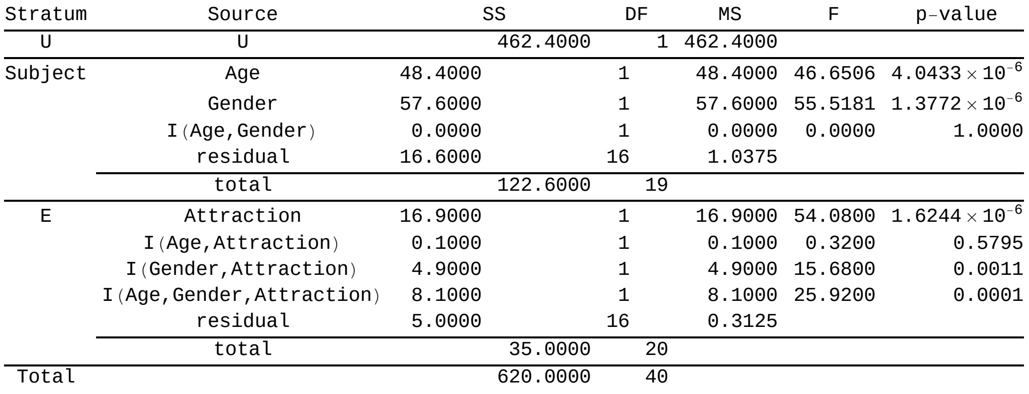


Fig. 3. Hasse diagram from *AutomaticAnova* for treatment factors in Example 2

read, but in general it is advisable to take such order effects into account.

The input file for the *AutomaticAnova* has forty rows and five columns. Of these three are for 'Age', 'Gender' and 'Attraction' with levels coded as 1 and 2 and one column 'y' contains the responses. Furthermore, there is a column called 'Subject' for a factor with levels $1, \ldots, 20$ representing the individual students. In addition to selecting the response variable, 'Subject' has to be specified as a plot factor and 'Age', 'Gender' and 'Attraction' as treatment factors. Since here all possible interactions of the treatment factors are of potential interest the default option for the level of interaction explained in Section 5.4 is used.



Stratum	Source	SS		DF	MS	F	p-value
U	U		462.4000	1	462.4000		
Subject	Age	48.4000		1	48.4000	46.6506	4.0433×10^{-6}
	Gender	57.6000		1	57.6000	55.5181	1.3772×10^{-6}
	I(Age,Gender)	0.0000		1	0.0000	0.0000	1.0000
	residual	16.6000		16	1.0375		
	total		122.6000	19			
E	Attraction	16.9000		1	16.9000	54.0800	1.6244×10^{-6}
	I(Age,Attraction)	0.1000		1	0.1000	0.3200	0.5795
	I(Gender,Attraction)	4.9000		1	4.9000	15.6800	0.0011
	I(Age,Gender,Attraction)	8.1000		1	8.1000	25.9200	0.0001
	residual	5.0000		16	0.3125		
	total		35.0000	20			
Total			620.0000	40			

Fig. 4. Anova table for Example 2 from *AutomaticAnova*

The Hasse diagram for the plot factors is not shown here: it is a chain similar to the one in Figure 1 with vertices for, from top to bottom, the universal factor U $(1, 1)$, 'Subject' $(20, 19)$ and the equality factor E $(40, 20)$, where for every factor F the pair (n_F, d_F) indicates the number of levels n_F and the degrees of freedom d_F. For the treatment factors the program produces the Hasse diagram in Figure 3 and the anova table in Figure 4 as explained in Section 2.2. In the table, a source label such as 'I(Age, Gender)' stands for the infimum of the two factors. The corresponding F test is for the interaction of 'Age' and 'Gender'. The sums of squares, degrees of freedom, mean squares and F tests in Figure 4 are identical to those reported by (Gamst et al., 2008, p. 393). □

6.2 Split-plot experiment

The next example considers a classical split-plot experiment. This type of design and the corresponding analysis are covered in many textbooks such as Cochran and Cox (1957), Montgomery (1991) and Sahai and Ageel (2000), among others.

Example 3 Littell et al. (2006, p. 135) describe a greenhouse experiment in which four doses of a pesticide were applied to two varieties of a plant. The experiment involved five benches which were used as blocks. Each bench was divided into four sections, each of which contained one plant of each variety. On each bench separately, the four doses were applied to the four sections in a random order. Although not mentioned explicitly, the authors' analysis is consistent with assuming that in each section the two varieties were also allocated at random. Thus, using common split-plot terminology, the sections represent whole plots and the plants are subplots. A response was measured on each plant and the data are available as Data Set 4.6 in Littell et al. (2006, Appendix 2).

Assuming that the columns in the input file are named 'Bench', 'Section', 'Dose', 'Variety' and 'y' for the response, when using the *AutomaticAnova* GUI the first two variables need to be specified as plot and the next two as treatment factors. A factor for the individual plants is not required, since this would coincide with the equality factor E on the set of the observational units, which is automatically included. In addition, 'y' has to be selected as the response variable. Since every bench has different sections, in the input file the coding convention in Section 5.2 needs to be observed, which implies that the factor 'Section' has twenty levels.

The Hasse diagrams, not shown here, look similar to those in Figure 1. For the plot factors the diagram is a chain with, from bottom to top, four vertices

```
Stratum     Source               SS           DF       MS         F    p-value
   U           U                  24319.6923      1 24319.6923
 Bench       Bench                  119.7340      4    29.9335
Section      Dose        545.5048              3      181.8349 13.6254  0.0004
            residual     160.1440             12       13.3453
             total                  705.6488     15
   E        Variety       11.9902              1       11.9902  2.7762  0.1151
        I(Variety,Dose)   29.6427              3        9.8809  2.2878  0.1176
            residual      69.1020             16        4.3189
             total                  110.7350     20
 Total                            25255.8100     40
```

Fig. 5. Anova table for Example 3 from *AutomaticAnova*

for E $(40, 20)$, 'Section' $(20, 15)$, 'Bench' $(5, 4)$ and U $(1, 1)$. For the treatment factors the Hasse diagram has a diamond shape with U $(1, 1)$ at the top, the infimum of 'Variety' and 'Dose' $(8, 3)$ at the bottom and separate vertices for 'Variety' $(2, 1)$ and 'Dose' $(4, 3)$ in between. By using the default setting in Section 5.4 for the level of interaction of the treatment factors, the anova table in Figure 5 is obtained. □

6.3 Latin square with randomized complete blocks in sub-columns

In the previous two examples the observational units had a very simple structure which could be represented by a chain in the Hasse diagram for the plot factors. A major strength of the *AutomaticAnova* is however that it can also handle situations with more complicated plot and/or treatment structures, as is illustrated next.

Example 4 The following description of an actual experiment together with an analysis of simulated data using SAS is presented by Federer and King (2007, pp. 47–51). The data are also available as a data set `sbex2_2` in a file `Example2.2.sas` which accompanies the book and these were used in the analysis reported below.

The experiment involves five apple tree rootstocks which were arranged in a 5 × 5 Latin square. In addition, four soil treatments were applied using a randomized complete block design where the five columns of the Latin square formed the blocks. Federer and King (2007, p. 47) state that "the rows and rootstocks of the Latin square are crossed with soil treatments" and later that the "soil treatments go across all rows" (p. 48). In order to better understand this description, first it should be noted that combinations of rows and rootstocks together identify the 25 different cells of the Latin square. Secondly, all four soil treatments are applied to every cell of the Latin Square which follows from the use of the word "crossing" in the above quote. Thirdly, that the soil treatments go across all rows together with the information that a randomized complete block design with columns of the Latin square as blocks was used

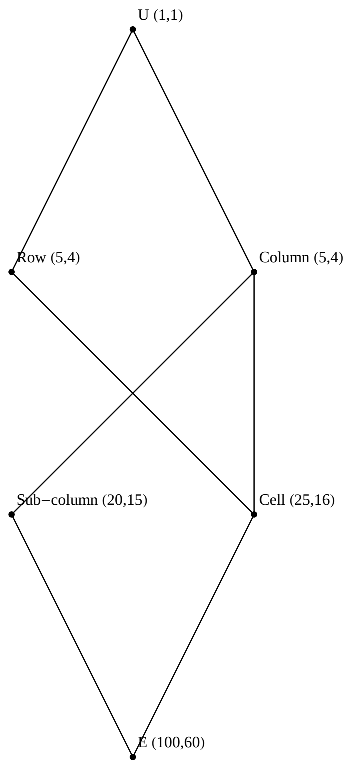


Fig. 6. Hasse diagram from *AutomaticAnova* for plot factors in Example 4

implies that effectively every column was divided into four sub-columns to which the soil treatments were applied in a random order.

In total there are $N = 100$ observational units, which are the intersections of the sub-columns with rows. The unit structure can be described by four plot factors 'Row', 'Column', 'Cell' and 'Sub-column' with 5, 5, 25 and 20 levels respectively, where the number of levels of the 'Sub-column' factor follows from the coding convention in Section 5.2. The treatment factors are 'Rootstock' with 5 and 'Soil tmt' with 4 levels.

Federer and King (2007) do not consider a separate factor 'Cell' but identify this with the effect for crossing 'Column' with 'Rootstock'. Likewise, instead of a factor 'Sub-column' they consider the crossing of 'Column' and 'Soil tmt'. In order to analyze the experiment, the authors need to perform the anova for two different models and combine these using some additional hand calculations, since "a SAS PROC GLM code for the appropriate analysis including the row blocking is not available" (Federer and King, 2007, p. 49). Also the error terms used in the F tests of the rootstock effect, the soil treatment effect and their interaction need to be indicated explicitly in additional SAS directives. These complications appear to be caused by not explicitly distinguishing between plot and treatment factors.

The *AutomaticAnova* with the default option "All interactions" on the other

Stratum	Source		SS		DF	MS	F	p-value
U	U		1.0002×10^8		1	1.0002×10^8		
Row	Row		147.4386		4	36.8596		
Column	Column		1318.6297		4	329.6574		
Sub-column	Soil tmt	351.8826		3		117.2942	0.3643	0.7800
	residual	3863.2837		12		321.9403		
	total		4215.1663		15			
Cell	Rootstock	1159.7645		4		289.9411	1.3074	0.3220
	residual	2661.1326		12		221.7611		
	total		3820.8971		16			
E	I(Soil tmt,Rootstock)	825.9588		12		68.8299	0.2757	0.9906
	residual	11982.2751		48		249.6307		
	total		12808.2338		60			
Total			1.0004×10^8		100			

Fig. 7. Anova table for Example 4 from *AutomaticAnova*

hand produces the combined analysis from an input file with columns 'Row', 'Column', 'Cell', 'Sub-column' 'Rootstock', 'Soil tmt' and 'y' for the response, where the levels of all factors are coded by the integers from 1 up to the number of factor levels, by only specifying which columns represent plot and treatment factors respectively and by selecting 'y' as the response variable. The Hasse diagram for the plot factors is shown in Figure 6. By swapping the vertices for 'Cell' and 'Sub-column' it is possible to obtain an equivalent version of the diagram in which no edges intersect. For the treatment factors the diagram is diamond shaped as in Figure 1 and hence not presented here. The anova table is shown in Figure 7. It should be noted that this analysis agrees with the combined analysis in Federer and King (2007, pp. 49–51) where it is described in the text only.

As an extension, Federer and King (2007, p. 99) consider the situation where the four soil treatments are the combinations of two factors 'Fumigation' and 'Composting' with two levels each and present the breakdown of the degrees of freedom in the corresponding anova. By including these factors into the

Stratum	Source	DF	
U	U		1
Row	Row		4
Column	Column		4
Sub-column	Fumigation	1	
	Composting	1	
	I(Fumigation,Composting)	1	
	residual	12	
	total		15
Cell	Rootstock	4	
	residual	12	
	total		16
E	I(Fumigation,Rootstock)	4	
	I(Composting,Rootstock)	4	
	I(Fumigation,Composting,Rootstock)	4	
	residual	48	
	total		60
Total			100

Fig. 8. Skeleton anova for extended version of Example 4 from *AutomaticAnova*

input file and re-running the *AutomaticAnova* with the same plot factors as before and treatment factors 'Rootstock', 'Fumigation' and 'Composting' and leaving the response variable unspecified, one obtains the skeleton anova table in Figure 8, which agrees with the results in Federer and King (2007, p. 99). □

6.4 Four treatment factors in nine strata

The final example to be considered in detail was presented by Professor T. Loughin at the Joint Statistical Meeting 2005 in Minneapolis and communicated to me by Professor C.-S. Cheng. It illustrates how by using the skeleton anova the correct analysis can be anticipated before data are available. The experimenters later decided to run a simplified version of the experiment and so there exist no response data for the original design described below (Loughin, personal communication).

Example 5 An experiment to determine what factors influence weed control and yield with genetically-altered soybean varieties involves $t = 126$ treatments which are all possible combinations of 3 varieties of soybean, 2 times of herbicide application, 3 rates of herbicide application and 7 weed species which were chosen based on resistance to a particular herbicide. Motivated by practical constraints a field is divided into 4 blocks, each of which can accommodate all 126 treatments in separate spots. In total there are then $N = 504$ spots which represent the observational units.

Every block is divided into 3 plots of equal size on which the soybean varieties are to be planted. Every such plot is divided further into two subplots to which the herbicide is applied early or late. In order to accommodate the different application rates every subplot is split again into three sub-subplots. Unrelated to the division into plots and smaller subunits each of the blocks is divided into 7 strips to which the weed species are allocated.

Figure 9 provides a schematic representation of the experimental layout. In this, the small rectangles represent the observational units or spots. Every block consists of 18 columns of 7 spots which are the sub-subplots to which one of the three application rates coded as 1, 2 or 3 is applied. Sets of three columns form the subplots and the time of herbicide application is shown above each subplot. Pairs of subplots make up the plots and the soybean variety to be planted on each plot is indicated. Finally, the weed species allocation to the strips or rows of each block is shown.

The input file for the *AutomaticAnova* has 504 rows for the observational units (plus one row for the column names) and nine columns. Five columns are for the plot factors B (blocks), P (plots), S (subplots), SS (sub-subplots)

Block 1

	Variety 1						Variety 3						Variety 2					
	Early			Late			Late			Early			Late			Early		
Weed 1 →	1	3	2	3	1	2	3	2	1	1	3	2	2	1	3	1	2	3
Weed 2 →	1	3	2	3	1	2	3	2	1	1	3	2	2	1	3	1	2	3
Weed 3 →	1	3	2	3	1	2	3	2	1	1	3	2	2	1	3	1	2	3
Weed 4 →	1	3	2	3	1	2	3	2	1	1	3	2	2	1	3	1	2	3
Weed 5 →	1	3	2	3	1	2	3	2	1	1	3	2	2	1	3	1	2	3
Weed 6 →	1	3	2	3	1	2	3	2	1	1	3	2	2	1	3	1	2	3
Weed 7 →	1	3	2	3	1	2	3	2	1	1	3	2	2	1	3	1	2	3

Block 2

	Variety 3						Variety 1						Variety 2					
	Late			Early			Early			Late			Early			Late		
Weed 3 →	2	1	3	3	2	1	1	2	3	3	1	2	3	2	1	2	1	3
Weed 4 →	2	1	3	3	2	1	1	2	3	3	1	2	3	2	1	2	1	3
Weed 7 →	2	1	3	3	2	1	1	2	3	3	1	2	3	2	1	2	1	3
Weed 5 →	2	1	3	3	2	1	1	2	3	3	1	2	3	2	1	2	1	3
Weed 2 →	2	1	3	3	2	1	1	2	3	3	1	2	3	2	1	2	1	3
Weed 1 →	2	1	3	3	2	1	1	2	3	3	1	2	3	2	1	2	1	3
Weed 6 →	2	1	3	3	2	1	1	2	3	3	1	2	3	2	1	2	1	3

Block 3

	Variety 1						Variety 2						Variety 3					
	Early			Late			Late			Early			Early			Late		
Weed 6 →	2	3	1	1	2	3	1	3	2	2	3	1	3	1	2	1	3	2
Weed 3 →	2	3	1	1	2	3	1	3	2	2	3	1	3	1	2	1	3	2
Weed 2 →	2	3	1	1	2	3	1	3	2	2	3	1	3	1	2	1	3	2
Weed 5 →	2	3	1	1	2	3	1	3	2	2	3	1	3	1	2	1	3	2
Weed 1 →	2	3	1	1	2	3	1	3	2	2	3	1	3	1	2	1	3	2
Weed 4 →	2	3	1	1	2	3	1	3	2	2	3	1	3	1	2	1	3	2
Weed 7 →	2	3	1	1	2	3	1	3	2	2	3	1	3	1	2	1	3	2

Block 4

	Variety 2						Variety 3						Variety 1					
	Late			Early			Early			Late			Late			Early		
Weed 2 →	2	1	3	3	1	2	2	3	1	1	2	3	3	2	1	2	3	1
Weed 4 →	2	1	3	3	1	2	2	3	1	1	2	3	3	2	1	2	3	1
Weed 5 →	2	1	3	3	1	2	2	3	1	1	2	3	3	2	1	2	3	1
Weed 7 →	2	1	3	3	1	2	2	3	1	1	2	3	3	2	1	2	3	1
Weed 6 →	2	1	3	3	1	2	2	3	1	1	2	3	3	2	1	2	3	1
Weed 3 →	2	1	3	3	1	2	2	3	1	1	2	3	3	2	1	2	3	1
Weed 1 →	2	1	3	3	1	2	2	3	1	1	2	3	3	2	1	2	3	1

Fig. 9. Experimental layout in Example 5

and ST (strips). A factor for the individual spots is not needed since this is equivalent to the equality factor E. The remaining four columns represent the treatment factors called 'Variety' with 3 levels, 'Time' with 2 levels, 'Rate' with 3 levels and 'Weed' with 7 levels. The coding convention in Section 5.2 implies that the plot factors have numbers of levels equal to $n_B = 4$, $n_P = 12$, $n_S = 24$, $n_{SS} = 72$ and $n_{ST} = 28$, which can be easily verified by looking at the experimental layout in Figure 9.

In the GUI, the plot and treatment factors need to be selected from the input spreadsheet. Since there is no response variable this remains unspecified. The experimenters were interested in testing all main effects and all possible interactions involving two, three or four of the treatment factors, which means that the default option "All interactions" in Section 5.4 can be used.

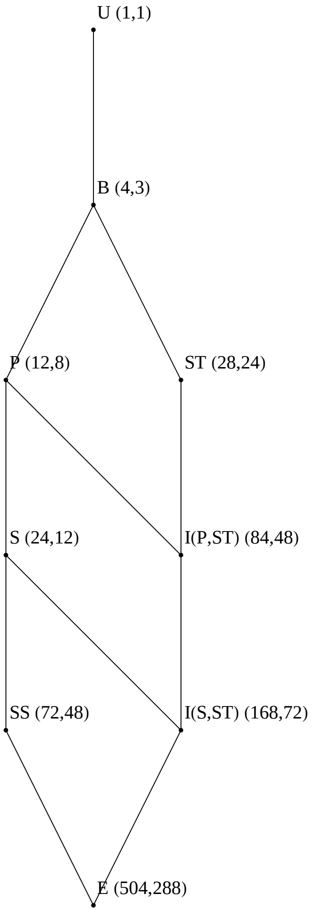


Fig. 10. Hasse diagram from *AutomaticAnova* for plot factors in Example 5

The Hasse diagram for the plot factors is shown in Figure 10. In addition to the original plot factors and E and U, the diagram contains two more infima. As explained in Section 2.2 every vertex in the diagram gives rise to one of the nine strata in the skeleton anova table in Figure 11. In the table, lines labeled by infima, such as ‘I(Time, Variety)’, of two or more treatment factors correspond to tests of interactions. Tests which are performed in a stratum with many residual degrees of freedom tend to have a higher power. □

6.5 Further examples

Additional examples, several of which are qualitatively different from the ones presented here, are reported in Bailey (2008, Chapter 10). For example, the strip-plot design in Bailey (2008, Fig. 10.32) illustrates a situation where the supremum of two plot factors ‘washing machine’ and ‘dryer’ needs to be included in the list of plot factors in order to obtain the correct analysis. The *AutomaticAnova* package handles this automatically even if ‘washing machine’ and ‘dryer’ are specified as the only plot factors. Similarly, in Example 10.21

Stratum	Source	DF	
U	U		1
B	B		3
P	Variety	2	
	residual	6	
	total		8
S	Time	1	
	I(Time,Variety)	2	
	residual	9	
	total		12
ST	Weed	6	
	residual	18	
	total		24
SS	Rate	2	
	I(Time,Rate)	2	
	I(Variety,Rate)	4	
	I(Time,Variety,Rate)	4	
	residual	36	
	total		48
I(P,ST)	I(Variety,Weed)	12	
	residual	36	
	total		48
I(S,ST)	I(Time,Weed)	6	
	I(Time,Variety,Weed)	12	
	residual	54	
	total		72
E	I(Rate,Weed)	12	
	I(Time,Rate,Weed)	12	
	I(Variety,Rate,Weed)	24	
	I(Time,Variety,Rate,Weed)	24	
	residual	216	
	total		288
Total			504

Fig. 11. Anova table for Example 5 from *AutomaticAnova*

the supremum of a plot factor 'cell' and the treatment factor 'T' needs to be included as a treatment pseudofactor 'P' and again this is automatically done by the software when 'column', 'block' and 'cell' are selected as the plot factors and 'T' is specified as the treatment factor. For each of the examples in Bailey (2008, Chapter 10) the reported analysis was replicated by using the GUI version of the *AutomaticAnova* package, which gave the same results in all cases.

As mentioned before, the orthogonal designs defined by Bailey (2008) represent a generalization of the simple orthogonal block structures in Nelder (1965a,b). Example 10 in Bailey (1991) describes a poset block structure which is not a simple orthogonal block structure. For this design the *AutomaticAnova* package also produces the correct analysis.

7 Discussion

At the start of the paper the question was raised whether another anova program is needed. In the previous sections I have tried to show that the *AutomaticAnova* package and its GUI offer a user-friendly alternative to standard software, which for a wide class of potentially complex designs can produce the correct analysis by asking the user only for minimal input information. The program's ability to deduce the analysis from the design and the specification of the plot and treatment factors alone is a unique feature which makes the specification of a model formula obsolete and thus to the best of my knowledge distinguishes the *AutomaticAnova* package from any existing anova procedures available in other software.

The usefulness of Hasse diagrams for representing anova models has been recognized by Taylor and Hilton (1981), Tjur (1984), Bergerud (1996) and Vilizzi (2005). Lohr (1995) emphasizes applications of the diagrams for teaching and consulting purposes. Several of these papers discuss how the diagram can be generated and used for deriving the anova table. However, none considers the question how the partial order underlying the diagram can be automatically extracted from the design.

The approach presented in the current paper also has its limitations and for the benefit of potential users, in particular non-statisticians, these should be pointed out very clearly. Firstly, the class of designs for which the *Automatic-Anova* package can be used does not include incomplete block designs nor augmented block designs in which new treatments are added to some but not all blocks, since both types of design violate some of the orthogonality requirements. Secondly, in order to be able to use the package, the design for the plot factors needs to be balanced in the sense of property (i)(a) in the definition of an orthogonal design. For the treatment factors this is not necessarily the case and Bailey (2008, p. 181) provides some examples of situations where the treatment factors are unbalanced. Yet, in general more often than not unbalanced designs are not amenable to an analysis by means of the methods in this paper. Thirdly, some designs confound effects of the treatment factors with effects of the plot factors and this can lead to violations of the orthogonality requirements. The *AutomaticAnova* package will however tell the user when such violations occur.

A different sort of limitation arises as a consequence of the balance between simplicity and comprehensiveness the program strives to achieve. More precisely, as explained in Section 5.4 the user can specify only the level of interaction of the treatment factors, but not, for example, that the analysis for a design with three treatment factors should only include one of the three possible two-factor interactions. When designing the package, offering this possibility by providing some additional options was ruled out, because it was deemed to make the program less usable for non-statisticians. Yet, a person versed in the anova may be able to work around similar problems by including additional factors representing specific contrasts in the input file.

The *AutomaticAnova* package has some additional features which space does not allow me to go into. For example, the program can also handle regular fractional factorial designs. In this regard, there are still some issues which concern the listing of all effects which are aliased with a given treatment effect which will hopefully be fully addressed in a later version of the package. Also, for the plot factors the *AutomaticAnova* output provides estimates of the variance components if response data are available or otherwise analytic formulae for the anova estimators of the variance components, which are derived from the design. Again, space does not allow me to go into any details except that the values in the output are estimates of the θ_F parameters in the proof of Theorem 10.9 in Bailey (2008).

Some extensions which are planned for the future are the addition of residual plots for checking model assumptions and the automatic suggestion of a Box-Cox transformation of the response variable where appropriate. At the moment the implementation has not been optimized for speed and, although usually the analysis only takes a few seconds, for some complex designs, such as the one in Example 5, the program may run for about one hour. It is hoped to also address this in a subsequent version.

The fact that the *AutomaticAnova* package has been developed for MATHEMATICA appears to be a hurdle for making the program available to a wide audience of users. The algorithms in Section 3 should be sufficient to enable others to implement versions of the program in other programming languages, although this may not be straightforward if the language does not support set operations. It is therefore planned to develop a web application which offers an interface to the *AutomaticAnova* functionality that can be used even without having access to MATHEMATICA.

Acknowledgements

The main part of this work was carried out at the Isaac Newton Institute (INI), Cambridge, during the ‘Design and Analysis of Experiments’ programme from July–December 2011. The author wishes to express his sincere gratitude to the INI and the programme organizers. I also thank Ching-Shui Cheng and Tom Loughin for providing information on the experiment in Example 5. Detailed comments by Rosemary Bailey on a draft of this paper are also much appreciated.

References

Bailey, R. A., 1981. A unified approach to design of experiments. J. R. Statist. Soc. A 144, 214–223.

Bailey, R. A., 1991. Strata for randomized experiments. J. R. Statist. Soc. B 53, 27–78.

Bailey, R. A., 1996. Orthogonal partitions in designed experiments. Designs, Codes and Cryptography 8, 45–77.

Bailey, R. A., 2008. Design of comparative experiments. Cambridge University Press, Cambridge, UK.

Bergerud, W. A., 1996. Displaying factor relationships in experiments. American Statistician 50, 228–233.

Christensen, R., 1987. Plane answers to complex questions. Springer, New York.

Cochran, W. G., Cox, G. M., 1957. Experimental designs, 2nd Edition. Wiley, New York.

Federer, W. T., King, F., 2007. Variations on split plot and split block experiment designs. Wiley, New York.

Gamst, G., Meyers, L. S., Guarino, A. J., 2008. Analysis of variance designs: a conceptual and computational approach with SPSS and SAS. Cambridge University Press, Cambridge, UK.

Gelman, A., 2005. Analysis of variance—why it is more important than ever. Annals of Statistics 33, 1–53.

Houtman, A. M., Speed, T. P., 1983. Balance in designed experiments with orthogonal block structure. Annals of Statistics 11, 1069–1085.

Littell, R. C., Milliken, G. A., Stroup, W. W., Wolfinger, R. D., Schabenberger, O., 2006. SAS for mixed models, 2nd Edition. SAS Institute Inc., Cary, NC.

Lohr, S. L., 1995. Hasse diagrams in statistical consulting and teaching. American Statistician 49, 376–381.

Montgomery, D. C., 1991. Design and analysis of experiments, 3rd Edition. Wiley, New York.

Muller, H., Großmann, H., Chittka, L., 2010. 'Personality' in bumblebees: individual consistency in responses to novel colours? Animal Behaviour 80, 1065–1074.

Nelder, J. A., 1965a. The analysis of randomized experiments with orthogonal block structure. I. Block structure and the null analysis of variance. Proceedings of the Royal Society of London, Series A 283, 147–162.

Nelder, J. A., 1965b. The analysis of randomized experiments with orthogonal block structure. II. Treatment structure and the general analysis of variance. Proceedings of the Royal Society of London, Series A 283, 163–178.

Payne, R. W., Wilkinson, G. N., 1977. A general algorithm for analysis of variance. Appl. Statist. 26, 251–260.

Sahai, H., Ageel, M. I., 2000. The analysis of variance: fixed, random and mixed models. Birkhäuser, Boston.

Speed, T. P., 1987. What is an analysis of variance? Annals of Statistics 15, 885–910.

Taylor, W. H., Hilton, H. G., 1981. A structure diagram symbolization for analysis of variance. American Statistican 35, 85–93.

Tjur, T., 1984. Analysis of variance models in orthogonal designs. International Statistical Review 52, 33–65.

Vilizzi, L., 2005. The linear model diagram: a graphical method for the display of factor relationships in experimental design. Ecological Modelling 184, 263–275.

Wilkinson, G. N., 1970. A general recursive procedure for analysis of variance. Biometrika 57, 19–46.

Wilkinson, G. N., Rogers, C. E., 1973. Symbolic description of factorial models for analysis of variance. J. R. Statist. Soc. C 22, 392–399.