Jour. Ind. Soc. Ag. Statistics
57 (Special Volume), 2004 : 320-344

Computer Aided Construction and Analysis of Augmented Designs

Abhishek Rathore[1], Rajender Parsad and V.K. Gupta
Indian Agricultural Statistics Research Institute, New Delhi

SUMMARY

The present investigation develops a β-version of Statistical Package for Augmented Design (SPAD). The package generates randomized layout of augmented designs and performs the analysis of data generated. For given number of test treatments (w), control treatments (u) and number of blocks (b), it computes the replication number of control treatment(s) in each block of the design such that the efficiency per observation of the test treatments vs control treatment(s) comparisons is maximum. The flexibility of choosing the replication number (r) of the control treatment(s) in each block is also provided. The randomized layout of the design is generated once the values of u, w, b are entered and r is chosen by the user. It also analyzes the data generated from the experiments conducted in augmented designs. A null hypothesis on any user-defined contrast can also be tested. The package is also very useful for classroom teaching as well as for the researchers in statistics with interest in experimental designs. The package has been developed using Microsoft Visual C++ 6.0.

Key words : Augmented block designs, Analysis of block designs, Contrast analysis, All possible paired treatment comparisons.

1. Introduction

In agricultural experiments often the existing practices called control treatments or check varieties are compared with new varieties or germplasms collected through exotic or domestic collections, called test treatments. In some cases experimental material for test treatments is limited and it is not possible to replicate them in the design. However, adequate material is available for replicating control treatments in the design. This kind of experimental situations came to be known to Federer [3] in screening new strains of sugarcane and soil fumigants used in pineapples. Augmented (Hoonuiaku) designs were introduced by Federer [4] to fill a need arising in screening new strains of sugarcane at "Experimental Station of Hawaiian Sugarcane Planters Association" on the basis of agronomic characters other than yield.

---

[1] Indian Institute of Soil Science, Bhopal

An augmented design is any standard design in control treatments augmented with additional (new or test) treatments in the complete or incomplete blocks in one-way heterogeneity setting and in the complete or incomplete rows and columns in 2-way heterogeneity set up. For one-way heterogeneity settings augmented block designs may have the blocks with unequal sizes. Augmented designs eliminating heterogeneity in two directions are called augmented row-column designs. Federer ([4], [5]) gave the analysis, randomization procedure and construction of these designs by adding the new treatments to the blocks of a randomized complete block (RCB) design and balanced lattice design. Federer [6] gave procedures useful for screening material inspection and allocation with a bibliography. Federer and Raghavarao [7] have given a general theory of augmented designs. They obtained augmented block designs using RCB designs and linked block designs and augmented row-column designs using a Youden square design. Federer *et al.* [8] gave general methods of construction of augmented row column designs. They also provided formulae for standard errors of estimable treatment contrasts. A survey of the literature reveals that generally such experiments are conducted using an augmented randomized complete block design.

For facilitating the experimenters in analyzing data generated from augmented RCB design, Agrawal and Sapra [1] developed a software AUGMENT 1. This software takes data in two text files and after processing the results, it places results in a text file. The user takes results from the output disk file. The limitation of this software is that it is DOS based and can analyze the data generated from an augmented RCB design only. In an augmented randomized complete block design, the test treatments are replicated once in the design and control treatments appear exactly once in each block. However, the experimenters often like to know how many times the control treatments be replicated in each block so as to maximize the efficiency per observation for making test treatments vs control treatments(s) comparisons? Parsad and Gupta [10] made an attempt to answer this question and obtained an expression for obtaining the replication number of control treatments in each block so that the efficiency per observation is maximized.

To make the exposition clear, consider an experimental situation in which $w = \sum_{j=1}^{b} k_j$ test treatments, which occur only once in the design, are to be compared with u control treatments each replicated r times in each of the b blocks such that j^{th} block size is $k_j + ur$, $\forall\ j = 1, \ldots, b$. The total number of experimental units is $w + ubr$ and the total number of treatments is w + u. If 1, ..., w denotes the test treatments and w + 1, ..., w + u the control treatments, then for maximizing the efficiency per observation, the number of times each control appears in each of the blocks is

$$r = \frac{\sqrt{u+b-1}\ \sqrt{w}}{ub} \tag{1.1}$$

provided $b + u - 1 \leq w$. For a single control situation, $u = 1$, and the expression (1.1) reduces to $r = \sqrt{\frac{w}{b}}$ provided $b \leq w$, which is always true.

There may, however, arise many combinations of w, u and b for which expression (1.1) does not yield a positive integer value of r. In such situations, a question that arises is as to what integer value of r should be taken. For this Parsad and Gupta [10] obtained efficiency per observation for $w \leq 100$, $b \leq 25$ and $u \leq 10$ such that $b + u - 1 \leq w$ and r was taken as $r^* = \text{int}(r)$ and $\text{int}(r) + 1$ in the expression (4.3) besides taking $r = 1$. A close scrutiny reveals that if value of $r >$ #. 42 then take $r^* = \text{int}(r) + 1$ and for values of r smaller than or equal to #. 42 take $r^* = \text{int}(r)$ for $u \geq 2$. For $u = 1$, the same rule applies but the value of r is taken as #. 45 instead of #. 42.

Once the expression for obtaining the number of replications of the control treatments has been obtained, the problem is to generate a randomized layout of the design followed by analysis of data generated. To assist the experimenter in planning, generating randomized layout of the design and analysis of experimental data, a user-friendly, menu driven, graphic user interface (GUI) based Statistical Package for Augmented Designs (SPAD) has been developed. Design and development of SPAD is discussed in Section 2 and Section 3 describes the features of SPAD.

2. *Design and Development of SPAD*

SPAD (Statistical Package for Augmented Design) has been implemented using object oriented programming language Microsoft Visual C++. The fundamental idea behind selection of an object-oriented language is its superior abilities for code reusability, portability and modular development. Additionally, Visual C++ is a powerful and fully equipped programming language for developing any kind of mathematical software. For detailed features of Visual C++, one may refer to Kruglinski [9], Young [11], Ellis and Strousstrup [2] and MSDN online library. Software is completely stand-alone and can be installed on any hardware platform with 32 Bit Microsoft Windows Operating System. Software can be executed with minimum specification of RAM for host Operating System. Installation of SPAD takes 2 MB of hard disk space and at least 1 MB free space for its working. Software is menu driven and very user friendly. It has a rich edit control for text editor and supports cut, copy, paste, undo, find and find-replace facilities. A Context Sensitive Help with Contents, Index and Search facilities are also available. The software is designed to assist experimenters in planning and analyzing augmented designs. Next section explains the working and features of SPAD.

3. *Features of SPAD*

Statistical Package for Augmented Design (SPAD) is capable of generating any augmented randomized complete block (RCB) design and augmented complete block design that have maximum efficiency per observation. It is also capable of analyzing data generated from augmented block designs (complete or incomplete). A window depicting the different features is given in Figure 3.1.1. These features are discussed in the sequel.

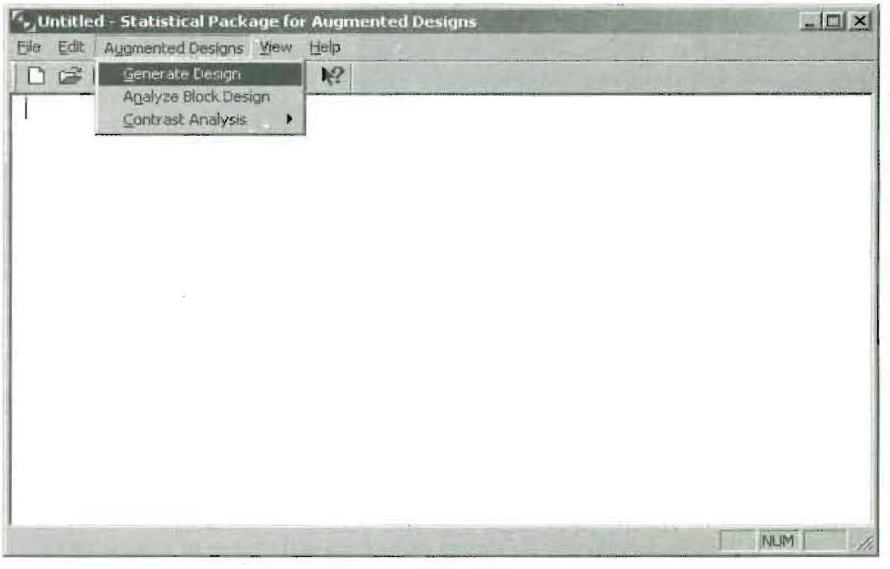


Fig. 3.1.1. Window depicting the features of SPAD

3.1 Generation of Augmented Designs

We begin with the generation of randomized layout of augmented complete block design with each control replicated a (≥ 1) times in each of the blocks. When a = 1, it reduces to usual augmented randomized complete block designs and when a = r as given in (1.1), then we get the randomized layout of the augmented complete block design that maximizes the efficiency per observation with respect to test treatments vs control treatment (s) contrasts. One can select the option Augmented Designs from the menu and then select the sub-option Generate Design (see Figure 3.1.1). On selecting the sub-option Generate Design, a form for entering the design parameters is displayed. For generation of randomized layout of augmented design, the input in terms of u (number of control treatments), w (number of test treatments) and b (blocks available with experimenter) is required. Once the user enters the design parameters, the replication of control treatment(s) that maximizes the efficiency per observation is automatically computed using (1.1) and suggested to the experimenter. There is flexibility for user to change the number of replications of the control(s) in

each of the blocks. To change replication of control treatment(s) one has to check on the "Change Replication of Control" check box. This will enable an edit box for replication of control treatment(s), where desired number of replication for control treatment(s) can be given (see Figure 3.1.2).

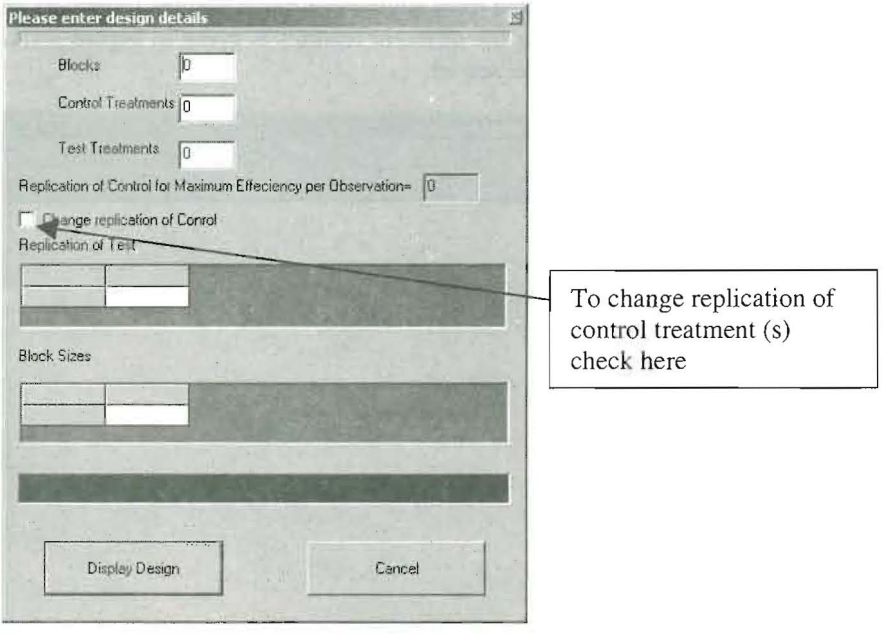


Fig. 3.1.2. Form for entering the design parameters and change of replication of control treatment(s)

Once the desired number of replications of control treatment(s) are entered, the box for entering replication of test treatments and block sizes get activated. Software also displays the total number of plots required. The block sizes are to be entered by the user. The block sizes should be such that $k_j \geq ua+1$ and $\sum_{j=1}^{b} k_j = n$, the total number of plots, $\forall j = 1,...,b$. A pseudo code for the generation of augmented block designs is given in Appendix-1. To make the exposition clear, we demonstrate with the help of following examples.

Example 3.1.1. Consider an experimental situation with u = 2, w = 8, b = 4 and k = 4. The optimum number of replications of the control treatment in each of the blocks obtained using (1.1) is 0.791 that is taken as one as per discussion after (1.1). Suppose that the user has selected the desired replication number of control treatment(s) in each of the blocks as one. Once, the choice of r is made, the box for entering the block sizes gets activated and total number of plots

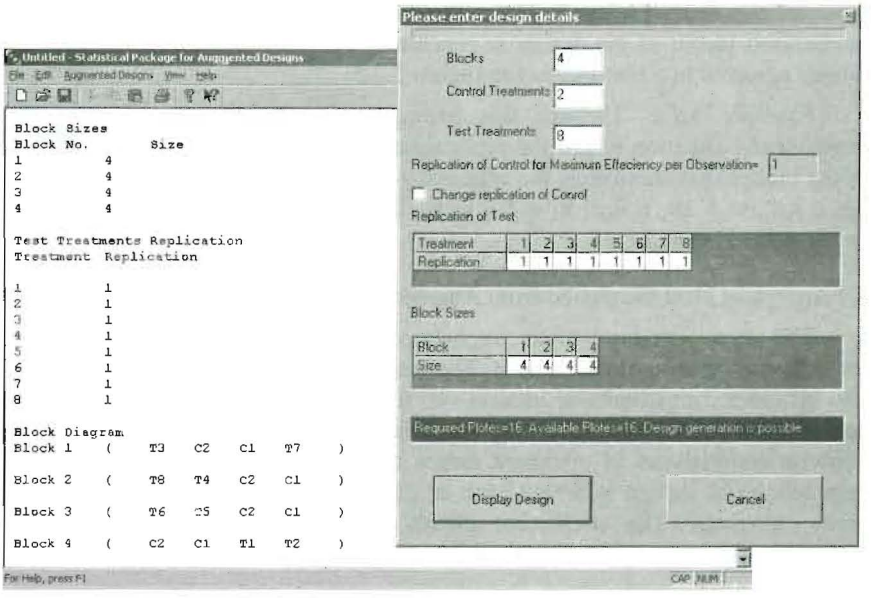


Fig. 3.1.3. Layout for the situation where control treatment (s) are replicated once in each block

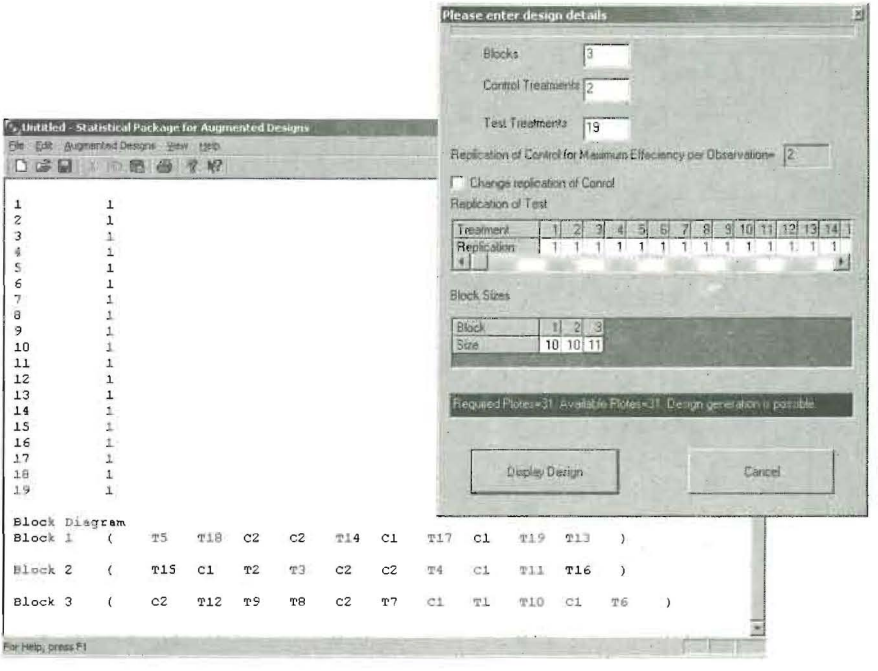


Fig. 3.1.4. Design layout for the situation where control treatment(s) are replicated twice in each block

required are also displayed. Once the blocks sizes are inputted, then the randomized layout of the design is generated. This randomized layout may be printed or saved in a file. Please see Figure 3.1.3.

Example 3.1.2. Through this example we illustrate the case of the experimental situation where the replication of the control treatment(s) in each of the blocks that maximizes the efficiency per observation is greater than one. Let u = 2, w = 19, b = 3, k_1 = 10, k_2 = 10, k_3 = 11. The results obtained are displayed in Figure 3.1.4.

3.2 Analyzing Data Generated from Augmented Design

The data pertaining to an augmented block design is analyzed as per procedure of general block designs. The treatment sum of squares is partitioned into different components of interest *viz.* (i) among test treatments, (ii) among control treatments and (iii) among test treatments and control treatments. The pairwise comparisons of treatment means can be simplified for an augmented complete block design in which each of the control treatments appear in each block 'a' times. The coefficient matrix of reduced normal equations for estimating treatment effects for the above design is

$$\mathbf{C}=\begin{bmatrix} ba\mathbf{I}_u-\left(a^2\sum_{j=1}^{b}\frac{1}{k_j+ua}\right)\mathbf{1}_u\mathbf{1}_u' & -\frac{a}{k_1+ua}\mathbf{1}_u\mathbf{1}_{k_1}' & -\frac{a}{k_2+ua}\mathbf{1}_u\mathbf{1}_{k_2}' & \cdots & -\frac{a}{k_b+ua}\mathbf{1}_u\mathbf{1}_{k_b}' \\ -\frac{a}{k_1+ua}\mathbf{1}_{k_1}\mathbf{1}_u' & \mathbf{A}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ -\frac{a}{k_2+ua}\mathbf{1}_{k_2}\mathbf{1}_u' & \mathbf{0} & \mathbf{A}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots\!\vdots & \vdots \\ -\frac{a}{k_b+ua}\mathbf{1}_{k_b}\mathbf{1}_u' & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A}_b \end{bmatrix}$$

where $\mathbf{A}_j=\mathbf{I}_{k_j}-\frac{1}{k_j+ua}\mathbf{1}_{k_j}\mathbf{1}_{k_j}'$; $\forall$ j = 1, ..., b. A generalized inverse of **C** is given by

$$\mathbf{C}^{-}=\begin{bmatrix} \frac{1}{ba}\left[\mathbf{I}_u-\frac{1}{u}\mathbf{1}_u\mathbf{1}_u'\right] & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{k_1}+\frac{1}{ua}\mathbf{1}_{k_1}\mathbf{1}_{k_1}' & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{k_2}+\frac{1}{ua}\mathbf{1}_{k_2}\mathbf{1}_{k_2}' & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots\!\vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I}_{k_b}+\frac{1}{ua}\mathbf{1}_{k_b}\mathbf{1}_{k_b}' \end{bmatrix}$$

The standard errors (S.E.) of the estimates of various elementary treatment contrasts are calculated as discussed in sequel.

(i) S.E. (Between two control treatments) = $\sqrt{\frac{2\sigma^2}{ba}}$

(ii) S.E. (between two test treatments in same block) = $\sqrt{2\sigma^2}$

(iii) S.E.(between two test treatments in different blocks) = $\sqrt{2\sigma^2(1+\frac{1}{ua})}$ and

(iv) S.E. (between a test treatment and a control treatment) $= \sqrt{\sigma^2\left(1+\frac{1}{ua}+\frac{1}{ba}-\frac{1}{uba}\right)}$

For an augmented incomplete block design, the significance of all possible paired treatment comparisons can be tested by automatically generating the all possible elementary treatment contrasts.

For performing the analysis of data generated through an augmented block design, an ASCII data file in a specified format, as described in sequel is required. The existing ASCII data file can be opened in the SPAD window using File-Open options. A new data file can also be created in the SPAD window using File-New option. One can also copy and paste data into SPAD editor from any windows based software like Excel or which supports clipboard operations. For creation of data file in a specified format, the treatments are renumbered as 1, 2, ..., u, u + 1, ..., u + w. Here first u treatments are the control treatment(s) and u + 1, ..., u + w are the test treatments. Data file contains at least three columns, first column represents block number, second column represents treatment number and third column consists of observed value of character. If there are more than one character to be analyzed, then these can be entered in fourth column onwards. There is no limitation on the number of characters present in the file. All these data values must be separated by a SPACE or a TAB.

For performing the analysis of the data generated through an augmented block design, one can select the sub-option *Analyze Block Design* from Option *Augmented Design* in the menu. A click on sub-option Analyze Block Design displays a dialog box. In this dialog box user must specify the character to analyze this time. This box will only appear if data file has more then one character. Once a character is selected for the analysis complete analysis with two ANOVA tables; one for testing the equality of treatment effects and another for testing the equality of block effects, R^2, Coefficient of Variation, Root Mean Square Error (RMSE), General Mean and adjusted treatment means is generated. For partitioning the treatment sum of squares into components of interest *viz.* (i) among test treatments, (ii) among control treatments and (iii) among test

treatments vs control treatments, one can select the sub-option *Contrast Analysis*. There are three options within the contrast analysis viz. (i) Augmented CB design, (ii) GBD for Tests vs Control(s) and (iii) User Defined Contrasts. Here Tests is used for test treatments and Control(s) for control treatments. If the data is generated from an augmented design in which each of the control treatments appear equally often in all the blocks, then the option *Augmented CB* design can be used for obtaining partitioned sum of squares and critical differences for performing all possible paired treatment comparisons. If the data is generated from an augmented incomplete block design, then the option *GBD for Tests* vs *Control(s)* may be used. In this option, the exact probability levels of significance of all possible paired treatment comparisons are given in a $(u + w) \times (u + w)$ matrix. A null hypothesis on any other contrast of interest can be tested using *User Defined Contrasts*. To make the exposition clear, we illustrate the analysis with the help of following examples.

Example 3.2.1. This example has been taken from Federer [4]. Four hypothetical check varieties (control treatments) are denoted by C1, C2, C3, C4 and eight hypothetical seedlings (new accessions) or test treatments are denoted by T1, T2, ..., T8 in 3 blocks of sizes 7, 6 and 7 respectively. The layout is given below.

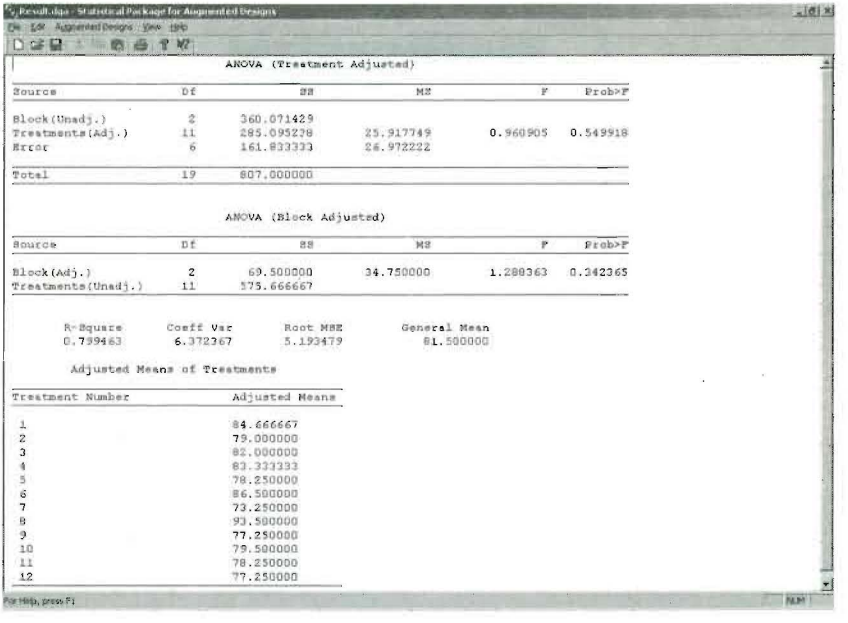
	Experimental Units						
	1	2	3	4	5	6	7
Block 1	T8 (74)	C3 (78)	C4 (78)	T3 (70)	C1 (83)	C2 (77)	T7 (75)
Block 2	C4 (91)	C2 (81)	C1 (79)	C3 (81)	T1 (79)	T5 (78)	
Block 3	T4 (96)	C3 (87)	C1 (92)	T2 (89)	C4 (81)	C2 (79)	T6 (82)

The values in the parenthesis are the observed values on the character under investigation.

The data file is created by renumbering the treatments from 1 to 12 where treatments 1, 2, 3, 4 are the control treatments and 5, 6, 7, 8, 9, 10, 11 and 12 are the test treatments. The format of the data file for the above example is given as

Block No.	Treatment No.	Observed Value.	Block No.	Treatment No.	Observed Value
1	1	83	2	4	91
1	2	77	2	5	79
1	3	78	2	9	78
1	4	78	3	1	92
1	7	70	3	2	79
1	11	75	3	3	87
1	12	74	3	4	81
2	1	79	3	8	96
2	2	81	3	6	89
2	3	81	3	10	82

Once data file is prepared and opened in the SPAD window, execute analysis module from menu by selecting Option Analyze Block Design. As there is only one character to be analyzed, therefore, a click on the Analyze Block Design option displays the analysis consisting of ANOVA tables (Block Adjusted and Treatments Adjusted), R^2, Coefficient of Variation, Root MSE, General Mean and adjusted treatment means (see Fig. 3.2.1.).



ANOVA (Treatment Adjusted)

Source	Df	SS	MS	F	Prob>F
Block(Unadj.)	2	360.071429			
Treatments(Adj.)	11	285.095238	25.917749	0.960905	0.549918
Error	6	161.833333	26.972222		
Total	19	807.000000			

ANOVA (Block Adjusted)

Source	Df	SS	MS	F	Prob>F
Block(Adj.)	2	69.500000	34.750000	1.288363	0.342365
Treatments(Unadj.)	11	575.666667			

R-Square	Coeff Var	Root MSE	General Mean
0.799463	6.372367	5.193479	81.500000

Adjusted Means of Treatments

Treatment Number	Adjusted Means
1	84.666667
2	79.000000
3	82.000000
4	83.333333
5	78.250000
6	86.500000
7	73.250000
8	93.500000
9	77.250000
10	79.500000
11	78.250000
12	77.250000

Fig. 3.2.1. Analysis results

Besides usual analysis of variance, experimenter is also interested in three types of contrasts namely, Among-Control (s), Among-Test(s) and Tests-vs-Control(s) Contrasts. On selecting the Contrast Analysis option "*Augmented CB Design*" user can see Sum of Squares, Mean square, F-Cal and Prob>F of these contrasts. Four Critical Differences (CD) are also be listed at 1% and 5% level of significance see Figure 3.2.2.

In addition to *Augmented CB Design*, SPAD can perform two more types of contrast analysis. These are *GBD for Tests vs Control(s)* and *User Defined Contrast*. GBD for Tests *vs* Control(s) option gives user an opportunity to carry out analysis of design for test *vs* control comparisons for the data generated from an augmented incomplete block design. Option *User Defined Contrasts* offers a facility to user to test any desired treatment contrast. On selecting *User Defined Contrast* option a form for taking contrasts appears see Figure 3.2.3.

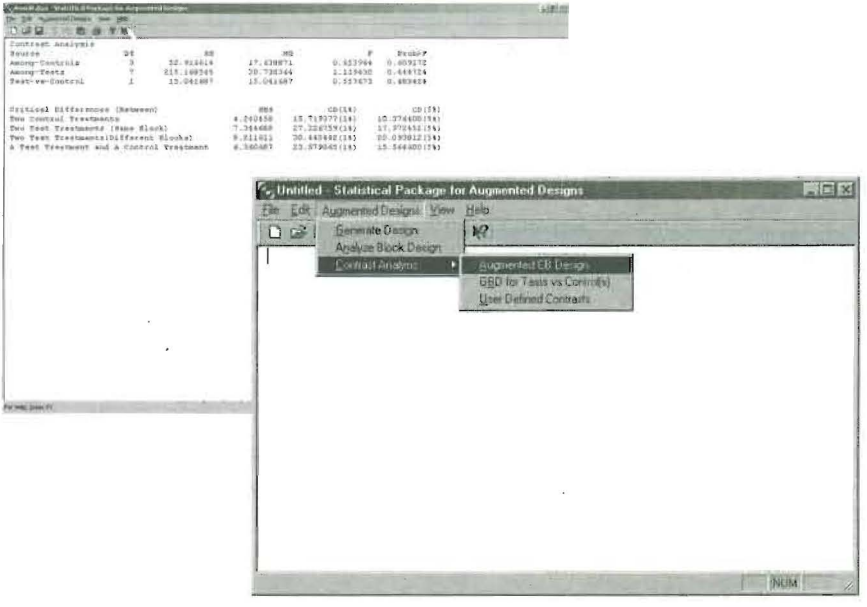


Fig 3.2.2. Results on selecting augmented CB design from contrast analysis option

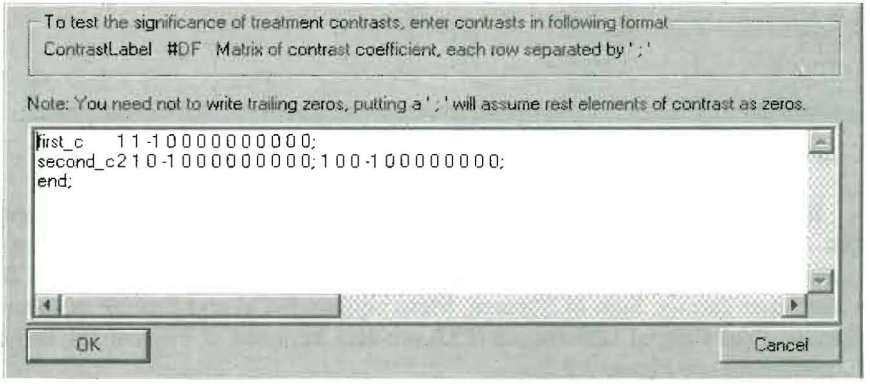


Fig. 3.2.3. User defined contrasts analysis

In this form user need to enter contrasts in a specific format, any number of contrasts can be tested simultaneously. To test a contrast, first enter its label, degree of freedom and then contrast coefficient matrix, each row separated by a semicolon ';'.

e.g.

first_c	1	1	-1	0	0	0	0	0	0	0	0	0	0;
second_c	2	1	0	-1	0	0	0	0	0	0	0	0	0;
	1	0	0	-1	0	0	0	0	0	0	0	0;	

end;

Here *first_c* is the name of contrast, with 1 degrees of freedom, where as *second_c* is another contrast with 2 degree of freedom. User can enter as many contrasts as desired. Each new contrast must start with a new line. After entering contrasts, declaration must be terminated by writing *end;* in the last line. Now a click on OK button will display result in SPAD window, see Figure 3.2.4.

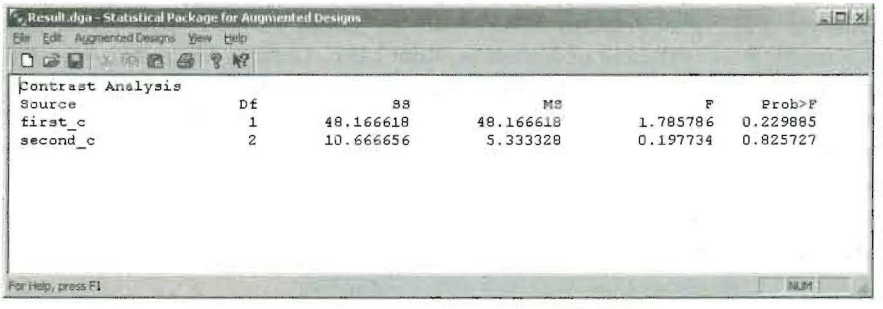


Fig. 3.2.4. Results of user defined contrasts analysis

Example 3.2.2. This illustration has been taken to demonstrate the analysis of augmented complete block design where each of the control treatment(s) is replicated in each of the block more than once. For this purpose, we modify the data given in illustration 1 by taking C1 and C2 as C1 and C3 and C4 as C2. Corresponding experimental situation will have parameters u = 2, w = 8, b = 3, $k_1 = 7$, $k_2 = 6$ and $k_3 = 7$ and will look like as given below.

Experimental Units

	1	2	3	4	5	6	7
Block 1	T8 (74)	C2 (78)	C2 (78)	T3 (70)	C1 (83)	C1 (77)	T7 (75)
Block 2	C2 (91)	C1 (81)	C1 (79)	C2 (81)	T1 (79)	T5 (78)	
Block 3	T4 (96)	C2 (87)	C1 (92)	T2 (89)	C2 (81)	C1 (79)	T6 (82)

On renumbering of treatments the data file will look like this.

Block No.	Treatment No.	Observed Value	Block No.	Treatment No.	Observed Value
1	1	83	2	2	91
1	1	77	2	3	79
1	2	78	2	7	78
1	2	78	3	1	92
1	5	70	3	1	79
1	9	75	3	2	87
1	10	74	3	2	81
2	1	79	3	6	96
2	1	81	3	4	89
2	2	81	3	8	82

To carry out this type of analysis first user the same steps as explained in Example 3.2.1. This displays ANOVA tables (Block Adjusted and Treatments Adjusted), R^2, Coefficient of Variation, Root MSE, General Mean and Adjusted treatment means. Four Critical Differences (CD) are listed at 1% and 5% level of significance. The exact probability levels of significance of all possible paired treatment comparisons can be obtained in a (u + w) × (u + w) matrix by selecting GBD for Tests vs Control(s) submenu from Contrast Analysis. The results obtained are displayed in Figures 3.2.5(a) and 3.2.5(b).

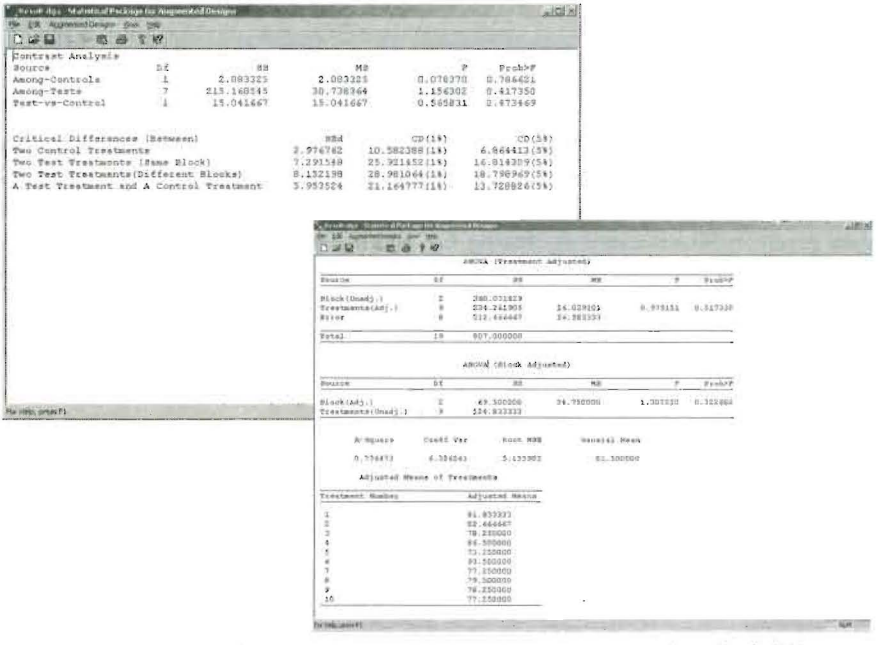
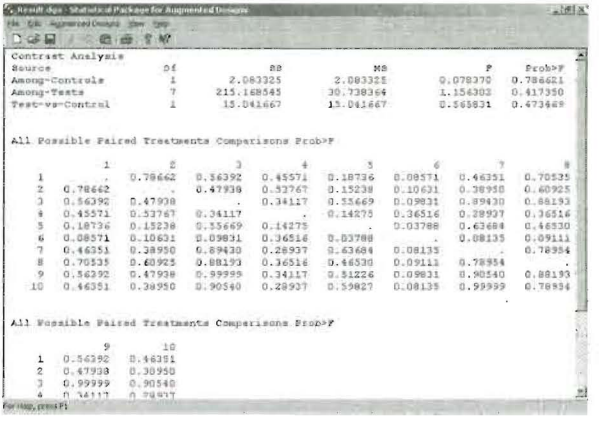


Fig. 3.2.5(a). ANOVA Tables, Adjusted treatment means and critical differences for Example 3.2.2



Contrast Analysis

Source	Df	SS	MS	F	Prob>F
Among-Controls	1	2.083325	2.083325	0.078370	0.786621
Among-Tests	7	215.168545	30.738364	1.156303	0.417350
Test-vs-Control	1	15.041667	15.041667	0.565831	0.473949

All Possible Paired Treatments Comparisons Prob>F

	1	2	3	4	5	6	7	8
1	.	0.78662	0.36392	0.45571	0.18736	0.08571	0.46351	0.70535
2	0.78662	.	0.47938	0.53767	0.15238	0.10631	0.38950	0.60925
3	0.36392	0.47938	.	0.34117	0.55669	0.09831	0.89430	0.88193
4	0.45571	0.53767	0.34117	.	0.14275	0.36516	0.28937	0.36516
5	0.18736	0.15238	0.55669	0.14275	.	0.03788	0.63684	0.46530
6	0.08571	0.10631	0.09831	0.36516	0.03788	.	0.08135	0.09111
7	0.46351	0.38950	0.89430	0.28937	0.63684	0.08135	.	0.78954
8	0.70535	0.60925	0.88193	0.36516	0.46530	0.09111	0.78954	.
9	0.56392	0.47938	0.99999	0.34117	0.51226	0.09831	0.90540	0.88193
10	0.46351	0.38950	0.90540	0.28937	0.59827	0.08135	0.99999	0.78954

All Possible Paired Treatments Comparisons Prob>F

	9	10
1	0.56392	0.46351
2	0.47938	0.38950
3	0.99999	0.90540
4	0.34117	0.28937

For Help, press F1

Fig. 3.2.5(b). Exact probability levels of significance matrix

Example 3.2.3. This illustration demonstrates the analysis of augmented incomplete block design. In the situation each block does not have all control treatment(s). For example consider an experiment with three blocks, 8 test treatments and 4 control treatments. Let block -1 contains control treatments 2, 3, 4, block-2 have 1, 3, 4 and block-3 have 1, 2 and 4 control treatments. This situation can be obtained by deleting control treatments 1, 2 and 3 from block 1, 2 and 3 respectively from data set used in Example 3.2.1. The parameters of the design are $u = 4$, $w = 8$, $b = 3$, $k_1 = 6$, $k_2 = 5$, $k_3 = 6$. The layout is given below

Experimental Units

	1	2	3	4	5	6
Block 1	T8 (74)	C3 (78)	C4 (78)	T3 (70)	C2 (77)	T7 (75)
Block 2	C4 (91)	C1 (79)	C3 (81)	T1 (79)	T5 (78)	
Block 3	T4 (96)	C1 (92)	T2 (89)	C4 (81)	C2 (79)	T6 (82)

On renumbering of treatments the data file will look like this.

Block No.	Treatment No.	Observed Value	Block No.	Treatment No.	Observed Value
1	2	77	2	5	79
1	3	78	2	9	78
1	4	78	3	1	92
1	7	70	3	2	79
1	11	75	3	4	81
1	12	74	3	8	96
2	1	79	3	6	89
2	3	81	3	10	82
2	4	91			

The analysis is carried out on the similar steps as in Example 3.2.2. For this experimental situation, the use of critical differences for paired treatment comparisons is not as simple as in case of augmented complete block designs. Therefore, the exact probability levels of significance of all possible paired treatment comparisons can be obtained in a (u + w) × (u + w) matrix by selecting GDB for Tests vs Control(s) submenu from Contrast Analysis Menu. Analysis Results are presented in Figure 3.2.6.

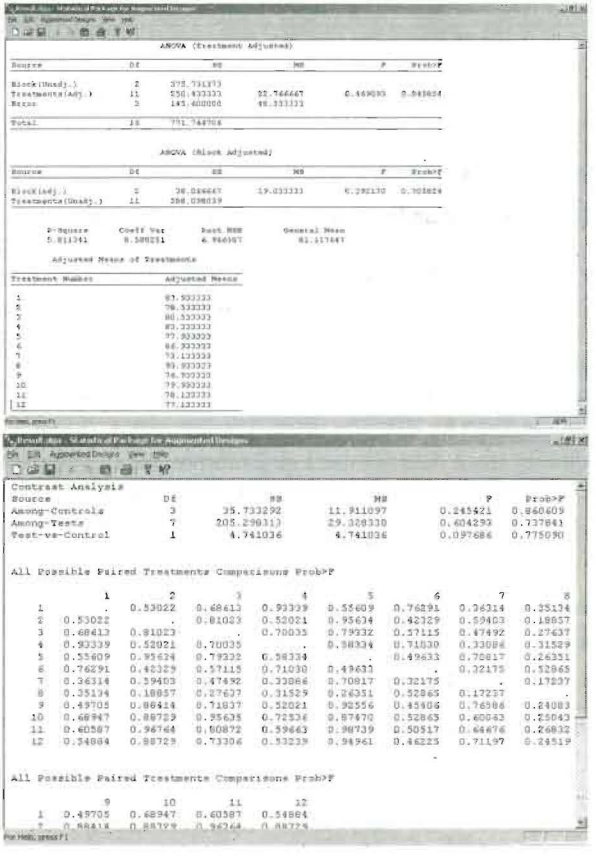


Fig. 3.2.6. Analysis results of an augmented incomplete block design

SPAD has facility to navigate between Data File, Analysis Results and Contrast Analysis Results from view menu. Alternatively one can use shortcut keys F4, F5 and F6 for these operations respectively. A psuedo code for the analysis of augmented designs is given in Appendix-2.

3.3 Context Sensitive Help

SPAD also has a context sensitive help, which provides detailed information on each feature of SPAD. Help module consists of information

about, how to start with SPAD, complete information about word processing capabilities of SPAD editor, how to Generate Augmented Block design, how to Analyze Block Design and also how to carry out Contrast Analysis etc. see Figure 3.3.1.

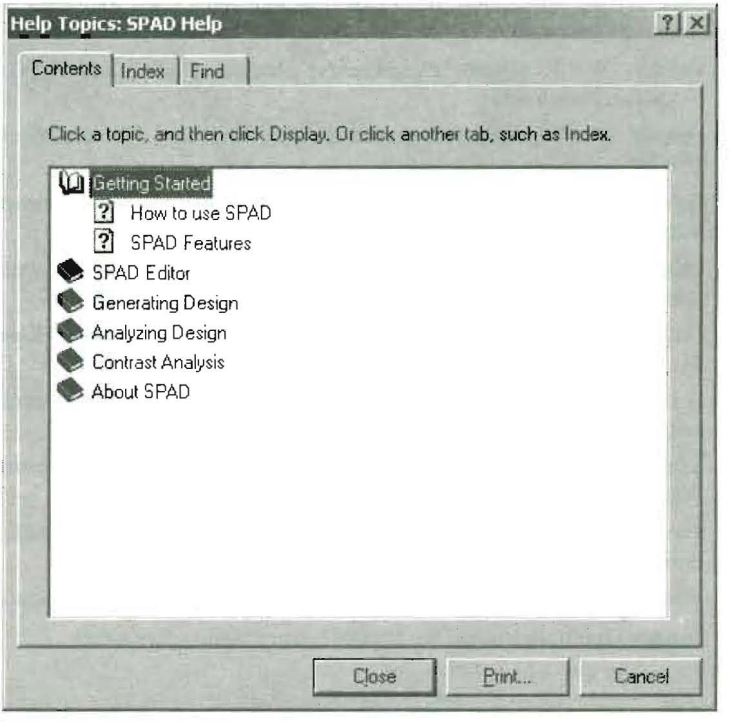


Fig. 3.3.1. SPAD help system

4. Discussion

This software is an effort to help the experimenters who are conducting experiments using augmented block designs. This software provides a complete solution in the sense that it is capable of generating the augmented complete block design that maximizes the efficiency per observation and analyzing the data generated from any augmented block design. In fact the data from any general block design can be analyzed using SPAD. Software is very easy to use and can be operated without any help or training. However, there is still a need to incorporate the features of analysis of covariance, stability analysis and variance components estimation from the data generated from an augmented block design.

REFERENCES

[1] Agrawal, R.C. and Sapra, R.L (1995). Augment 1: A microcomputer based program to analyze augmented randomized complete block design. *Indian J. Pl. Genet. Resources*, **8(1)**, 63-69.

[2] Ellis, M. and Strousstrup, B. (1990). *The Annotated C++ Reference Manual.* Addition Wesley, MA.

[3] Federer, W. T. (1955). *Experimental Design: Theory and Applications.* Mcmillan, New York.

[4] Federer, W. T. (1956). Augmented designs. *Hawaiian Planters Record*, **55**, 191-208.

[5] Federer, W. T. (1961). Augmented designs with one way elimination of heterogeneity. *Biometrics*, **17**, 447-473.

[6] Federer, W. T. (1963). Procedures and designs useful for screening material in selection and allocation with a bibliography. *Biometrics*, **19**, 553-587.

[7] Federer, W. T. and Raghavarao, D. (1975). On augmented designs. *Biometrics*, **31**, 39-45.

[8] Federer, W. T., Nair, R.C. and Raghavarao, D. (1975). Some augmented row column designs. *Biometrics*, **31**, 361-374.

[9] Kruglinski, D.J. (1997). *Inside Visual C++*. Fourth Edition. Microsoft Press, Washington.

[10] Parsad, R. and Gupta, V.K. (2000). A note on augmented designs. *Indian J. Pl. Genet. Resources*, **13(1)**, 53-58.

[11] Young, M.J. (1998). *Mastering Visual C++ 6.0.* BPB Publication, New Delhi.

[12] January (2000). Release of the MSDN Library, Online Resources.

APPENDIX 1

Pseudo code for generation of augmented complete block designs in which each of the 'u' controls are replicated r (≥1) times in each of the blocks.

m_B = Blocks = Number of Blocks

m_CT = Number of Control treatments

m_TT = Number of Test Treatments

Read(m_B, m_CT, m_TT)

BlkV[Blocks] = An Array Loop holding Size of Blocks, its Dimension should be equal to number of blocks

```
//Now compute replication of control treatment for maximum efficiency
//per observation
//Check whether m_b+m_CT-1 is less than or equal to m_TT

Treatments=m_TT;
CT=m_CT;
DecisionNo = sqrt( (CT+Blocks-1)*Treatments)/(CT*Blocks)
TempNo=DecisionNo-int(DecisionNo);
        if(CT=1)
        {
                if(TempNo>0.45)
                        RepC=int(DecisionNo)+1;
                else
                        RepC=int(DecisionNo);
        }
        else
        {
                if(TempNo>0.42)
                        RepC=int(DecisionNo)+1;
                else
                        RepC=int(DecisionNo);
        }
//If calculated Replication is less then one and greater than 0.42(0.45) make it one.
        if(RepC<1)RepC=1;
        m_RCT = A variable to hold calculated replications of control
        m_RCT=RepC;
//Since user can also specify replication of control, so check for
//possible error
        if(m_RCT<=0)
        {
        Show Error("Replication of Control Treatment can not be zero")
        And Return.
        }
```

```
//Now read all Block Sizes
  Loop (int i=0; i<Blocks; ++i)//i as a Loop Index Variable
  {
                Read BlkV[i]
//Now Check that Block Size is greater than the product of number of control treatments
and replications of the control treatments in each of the blocks
        if( BlkV[i] < CT*RepC+1 )
        {
        Show Error("Block is too small to hold even a single test treatment with all
        control treatments")
        And Return.
        }
        if(BlkV[i]==0)
        {
        Show Error(Block size can not be zero)'
        And Return.
        }
  }
RepV[Treatments] = An Array for holding Number of Test Treatments, its Dimension
should be equal to number of Test Treatments.
        Loop(i=0;i<Treatments;++i)
        {
                Read RepV[i]
                if(RepV[i]==0)
                {
                Show Error("Replication of Test Treatment can not be zero")
                And Return.
                }
        }
        m_RCT = Replication of Control Treatment
TotalBSize=Variable to hold total plots available in experiment
TotalPlotsReq= Variable equal to plots required in experiment
//Initialize them to zero
        TotalBSize=0
        TotalPlotsReq=0
//Add all block sizes to know total Plots available in field
        Loop(i=0;i<Blocks;++i)
                TotalBSize+=BlkV[i];
//Add Replications of test treatments to know total Plots required for them
        Loop(i=0;i<Treatments;++i)
                TotalPlotsReq += RepV[i];
```

```
//Now Add Plots required for Control treatments also
        TotalPlotsReq += CT*RepC*Blocks;
//Now Check whether available plots are sufficient OR not
        if(TotalPlotsReq > TotalBSize)
        {
        Show Error("Insufficient Plots")
        And Return.
        }
MaxBSize = Variable to hold plots available in largest block
        MaxBSize=0;
        flag=0;
        int Range;
// Now Find out maximum Block Size available
        Loop(int i=0;i<Blocks;++i)
        {
                if( BlkV[i] > MaxBSize) MaxBSize=BlkV[i];
        }
        RemainingT=0;
//Collecting Total Test Treatments Involved
        Loop(i=0;i<Treatments;++i)
                RemainingT+=RepV[i];
//Allcoate Space for Test and Control Treatments Location.
OTL = An two dimensional array which is going to hold design
        OTL [Blocks] [MaxBSize];

//Set All elements of Array OTL to -1, which will be a notation for free
//plots.
        Loop( i=0;i<Blocks;++i)
                Loop( j=0;j<MaxBSize;++j)
                        OTL[i][j]=-1;
//      Select Location (Randomly) for all Control Treatments
//      Select one plot in a block randomly and assign treatment to it
//      Select a block and select a control treatment
        Loop(i=0; i<Blocks; ++i)
        {
                Loop(int j=0;j<CT;++j)
                {
//Now, start a loop for all replications of Control treatments
                Loop(int r=0;r<RepC;++r)
                {
//Start an unending loop, till a replication of control is assigned to //any plot
```

```
                        while( 1 )
                        {
                        MakeMeLive();
//Select a plot in selected block randomly
                        RNO=random( BlkV[i]);
//Check whether selected plot is free or not, -1 will reflect free
//plot. If plot is free no more trials, quit from unending loop
        if( OTL[i][RNO]==-1)break;
//If plot was not free, continue with selection of new plot
        }
//Now, once a free plot is achieved, assign a control to it.
//Assignment refers to increase value of OTL array by number of
//control treatment + 2000. Here 2000 is an outstanding number refers
//to control treatment.
        OTL[i][RNO] = j+2000;
                        }
                }
        }
//Once all controls are assigned, start assignments of test treatments
//randomly Select Location (Randomly) for all Test Treatments. Start a
//loop, till all test treatments are assigned. Variable RemainingT will
//keep count of remaining test treatments. We have initialize RemainingT
//with total test treatments.
        while(RemainingT)
        {
//Select any test treatment randomly.
        RTN = random(Treatments);
//Now check, whether, that test treatment has some unassigned
//replications. i.e, Select Treatment whose Replication is Non Zero
        while( ! RepV[RTN] )
        {
//If first selected control treatment have already exhausted, select any other test
//treatment randomly, and repeat these steps till an //unassigned test treatment is attained.
        RTN = random(Treatments);
        }
//If a test treatment is attained, reduce its replication by one
//i.e One Replication of test treatment RTN is allotted
        --RepV[RTN];
//Till here a test treatment has been selected randomly, now select a //block for this test
treatment randomly
RBN = Variable used to keep randomly selected block number
flag =0;//This flag will insure a free plot has been found
```

```
        while(! ( (! Exist( RTN, RBN ) ) && (flag==1) ) )
        {
//Check whether Treatment appears in Block or Not, If appears then search
//Another Block Randomly.
                RBN = random( Blocks );
                flag=0;
//Now check whether block is having any free location or Not.
                Loop(int i_b=0;i_b<BlkV[RBN];++i_b)
                {
                        if( OTL[RBN][i_b]==-1)
                        {
                        flag=1;
                        break;
                        }
                }
        }
//Now Select any Free slot available in Block RBN
                while( Occupied( random( BlkV[RBN]), RBN ) );
//Till here we have got one randomly selected free plot in a randomly
//selected block for a randomly selected test treatment. Assign it.
OTL[RBN][RandomNO]=RTN;
//Decrease selected test treatment replication by one
                --RemainingT;//One treatment is Allotted
        }
//Now Display the Design
}
```

APPENDIX 2

Pseudo Code for Analysis of Data

Open Data File
Determine Number of characters and ask from user the character to be analyzed
Determine v, b *and* TotalObs
    v = Number of Treatments
    b = Number of Blocks
    TotalObs denotes total number of observations
Create a Matrix $N_{(v \times b)}$, Incidence Matrix
Create a Matrix $R_{(v \times b)}$, A diagonal Matrix of Replication Number
Create a Matrix $K_{(v \times b)}$, Block Matrix of Block Sizes
GSum is a variable to hold grand sum
Set GSum=0
Loop 1 to Total Observations
    Read BNO(Block No), TNO(Treatment No), OBS(Observation)
    Set N [TNO-1][BNO-1] = N [TNO-1][BNO-1]+1
    Set R [TNO-1][TNO-1] = R [TNO-1][TNO-1]+1
    Set K [BNO-1][BNO-1] = K [BNO-1][BNO-1]+1
    Set GSum = GSum + OBS
End Loop
CF is correction factor
Set CF=(Gsum*Gsum)/ TotalObs

Calculate C and C inverse
Set C = R – N * Inverse (K) * Transpose(N)
Set C = C + 1.0
Set DetValue = GetDeterminantValue (C)
if (DetValue = 0)
Disconnected design, Further Analysis will not be done.
End Analysis
Set C = Inverse(C)
Calculate Vector of Adjusted Treatment Total
Set Q = T – N * K * B
For ANOVA Treatments Adjusted
Set Treatment SS = Transpose (Q) * C * Q
Set Treatment DF = v - 1

Set Block SS = Transpose (B) * K * B - CF
Set Block DF = b - 1
Set Total SS = TotalSS - CF
Set Total DF = TotalObs - 1

Set Error SS = Total SS - Treatment SS - Block SS

Set Error DF = Total DF - Treatment DF - Block DF

For ANOVA Block Adjusted

Set Adjusted Treatment SS = Transpose (T) * R * T - CF

Set Adjusted Treatment DF = v - 1

Set Adjusted Block SS = Total SS – Error SS - Adjusted Treatment SS

Set Adjusted Block DF = b - 1

Now prepare and display ANOVA Tables

Contrast Analysis

1. Augmented CB Design

First Read number of Treatments to Prompt to User

u=controls, w=tests

u = Selected controls treatments

w = TotalT - u

Among Control Contrasts

Create a Matrix $PC_{(u-1,\ u+w)}$, a complete of linearly independent contrasts among control treatments.

```
Loop i = 0, i < u – 1, ++i
        PC [i][0] = 1;
        PC [i][i+1] = – (i+1)
        Loop j=1;j<i+1 ; ++j
                PC [i][j]=1;
        End Loop
End Loop
```

Among Test Contrasts

Create a Matrix $PT_{(w-1,\ u+w)}$ a complete of linearly independent contrasts among test treatments.

```
Loop    i=0, i<w – 1, ++i
        PT [i][u]=1
        PT [i][u+i+1]=-(i+1)
        Loop    j=1, j<i+1, ++j
                PT [i][u+j]=1
        End Loop
End Loop
```

Test *vs* Control Contrasts

Create a Matrix $PCT_{(1,\ u+v)}$ a test treatments vs control treatment(s) contrast

Set all elements of PCT = –u

```
        Loop i=0, i<u, ++i
        PCT [0][i]=w
End Loop
```

For Among-Controls, For Among-Tests, For Tests-vs-Control(s)

Take P = PC, PT and PCT one by one and calculate steps below

TempMat = P * Inverse(C) * Q

ContSumSQ = (Transpose(TempMat) * Inverse((P * Inverse(C) * Transpose(P)) * TempMat)

ContProb = GenFProb((ContSumSQ/ContDF)/(Error SS / Error DF), Contrast DF, Error DF)

Compute Critical Differences at 1% and 5% level of significance for comparing the means of control treatments, two test treatments in the same block, two test treatments in the different blocks AND a test treatment and a control treatment.

Display Results ContName ContDF ContSumSQ ContProb

2. GBD for Tests vs Control(s)

Do the Augmented CB Design Analysis

Calculate All Possible Paired Treatments Comparisons Prob>F

Generate All Possible Elementary Contrasts and Pace them in a Matrix PA, Total = ${}^{v}C_{2}$

ContDF = Degree of freedom of Contrast = 1

Loop i=0, i<u, ++i

Take First Row of PA and Say P

TempMat = P * Inverse(C) * Q

ContSumSQ = (Transpose(TempMat) * Inverse((P * Inverse(C) * Transpose(P)) * TempMat)

ContProb = GenFProb((ContSumSQ/ContDF)/(Error SS / Error DF), Contrast DF, Error DF)

Store in A Matrix

End Loop

Display Results in Matrix

3. User Defined Contrast

Take P = Matrix of contrast coefficients. Take one by one all entered contrasts

ContDF = Degree of freedom of Contrast

TempMat = P * Inverse(C) * Q

ContSumSQ = (Transpose(TempMat) * Inverse((P * Inverse(C) *

Transpose(P)) * TempMat)

ContProb = GenFProb((ContSumSQ/ContDF)/(Error SS / Error DF),

Contrast DF, Error DF)

Display Results ContName ContDF ContSumSQ ContProb

Term Matrix refers to a Two Dimensional Array

Inverse ( ) is a function to obtain true inverse of a given matrix

Transpose ( ) is a function to obtain Transpose of a given matrix

GetDeterminantValue( ) is a function to compute determinant value of a matrix

GenFProb() Calculates Prob>F

Algorithm and source code for above functions are available with the Author