

The %MktRuns Macro

Introduction

The %MktRuns autocall macro suggests reasonable sizes for experimental designs. It tries to find sizes in which perfect balance and orthogonality can occur, or at least sizes in which violations of orthogonality and balance are minimized.

%MktRuns Macro Syntax

%MktRuns(*list* <, *optional arguments*>)

Required Argument

list

specifies a list of the numbers of levels of all the factors. For example, for three 2-level factors, specify either `2 2 2` or `2 ** 3`. Lists of numbers like `2 2 3 3 4 4` or a *levels**number of factors* syntax like `2**2 3**2 4**2` can be used, or both can be combined: `2 2 3**4 5 6`. The specification `3**4` means four 3-level factors. The factor list is a positional parameter. This means that it must come first, and unlike all other parameters, it is not specified after a name and an equal sign.

Optional Arguments

INTERACT=*interaction-list*

specifies interactions that must be estimable. By default, no interactions are guaranteed to be estimable. For example:

`interact=x1*x2`

`interact=x1*x2 x3*x4*x5`

`interact=x1|x2|x3|x4|x5@2`

The interaction syntax is similar to that of PROC GLM and many of the other SAS/STAT modeling procedures. It uses ‘*’ for simple interactions (`x1*x2` is the interaction between `x1` and `x2`), ‘|’ for main effects and interactions (`x1|x2|x3` is the same as `x1 x2 x1*x2 x3 x1*x3 x2*x3 x1*x2*x3`), and ‘@’ to eliminate higher-order interactions (`x1|x2|x3@2` eliminates `x1*x2*x3` and is the same as `x1 x2 x1*x2 x3 x1*x3 x2*x3`). The specification ‘@2’ creates main effects and two-way interactions. In contrast with the syntax of PROC GLM, some shortcuts are permitted. For the factor names, you can either specify the actual variable names (for example, `x1*x2 . . .`) or specify only the factor number without the ‘x’ (for example, `1*2`). You can also specify `INTERACT=@2` for all main effects and two-way interactions, omitting the `1|2| . . .`. The following three specifications are equivalent:

```
%mktruns(2 ** 5, interact=@2)
%mktruns(2 ** 5, interact=1|2|3|4|5@2)
%mktruns(2 ** 5, interact=x1|x2|x3|x4|x5@2)
```

MAX= n <M> <F>

specifies the maximum number of design sizes to try. By default, MAX=200 2 F. The macro tries up to n sizes, starting with the saturated design. The macro stops trying larger sizes when it finds a design size with zero violations that is m times as big as a previously found size with zero violations. The macro reports the best 10 sizes. For example, if the saturated design has 10 runs, and there are zero violations in 16 runs, then by default, the largest size that the macro will consider is $32 = 2 \times 16$ runs. A third optional value of 'f' or 'F' is specified by default, which instructs the %MktRuns macro to not consider designs larger than full-factorials. If you specify MAX= without this option, larger designs might be considered.

MAXLEV= n

specifies the maximum number of levels to consider when the orthogonal array list is generated. By default, MAXLEV=50, and the actual maximum is the max of the specified MAXLEV= value and the maximum number of levels in the factor list. Specify a value $2 \leq n \leq 144$.

MAXOA= n

specifies the maximum number of orthogonal arrays to display in the list of orthogonal arrays that the %MktEx macro knows how to make. By default, when no value is specified, the entire list is displayed. For example, you can specify MAXOA=0, OUTORTH=OA to get a data set instead of displaying this table. You could instead specify MAXOA=10 to see only the first 10 arrays.

N= n

specifies the design size to evaluate. By default, this option is not specified, and the MAX= argument specification provides a range of design sizes to evaluate.

OPTIONS=options-list

specifies binary options. You can specify one the following values:

JUSTPARSE

is used by other %Mkt macros to have this macro parse only the list argument and return it as a simple list of integers.

MULTIPLE

specifies that a term that is required for orthogonality can be counted multiple times when orthogonality violations are being counted. For example, combinations of levels for the pair of variable 1 and variable 2 must have equal frequencies for orthogonality in the main effects, and if two-way interactions are also required, then the (1, 2) pair must have equal frequencies again. By default, each combination of variables is counted only once.

The difference between the single and multiple sources is that the single-source method counts only the places in the design where equal frequencies must occur. The multiple-source method weights this count by the number of times each of these terms is important for achieving orthogonality. The results of the two methods are often highly correlated, but they can differ.

MULTIPLE2

specifies both `OPTION=MULTIPLE` and more detailed output, including the reason why each term is added to the source table when `OPTIONS=SOURCE MULTIPLE2` is specified.

NOPRINT

suppresses the display of all output. The only output from the macro is the output data sets.

NOSAT

suppresses inclusion of the saturated design in the design list. By default, the saturated design is reported along with the reasonable design sizes, even if it is not a very reasonable size, in order to provide some context for the other numbers. If you specify `OPTIONS=NOSAT`, then the saturated design is added to the list only if it meets the usual criteria for being a reasonable size.

SOURCE

displays the source of all the numbers in the table of reasonable design sizes.

512

adds to the catalog some larger designs in 512 runs with mixes of 16-, 8-, 4-, and 2-level factors, giving added flexibility in 512 runs at a cost of much slower run time. This option replaces the default $4^{160}32^1$ parent with $16^{32}32^1$.

By default, none of these options are specified.

OUT=SAS-data-set

specifies the name of a SAS data set that contains the suggested sizes. By default, `OUT=Nums`.

OUTORTH=SAS-data-set

specifies the output data set that contains the orthogonal array list. By default, this data set is not created.

TOOBIG=*n*

stops problems that are too big and take too long before they consume large quantities of resources. By default, `TOOBIG=80200`. The value is the maximum number of levels and pairs of levels that are considered. For models that contain interactions, the interaction terms also contribute to this value. A main-effects model that has 400 factors is small enough ($400 \times 401/2 = 80200$) to work with the default `TOOBIG=80200`. If you want to use the %MktRuns macro with more factors, you must specify a larger value for `TOOBIG=`. The following steps do not require you to specify the `TOOBIG=` argument:

```
%mktruns(2 ** 400)
%mktruns(2 ** 27, interact=@2)
```

The following steps work with the `TOOBIG=` argument specified, but they are too large to work without it being specified:

```
%mktruns(2 ** 401, toobig=80601)
%mktruns(2 ** 28, interact=@2, toobig=81865)
```

Help Option

You can specify either of the following to display the option names and simple examples of the macro syntax:

```
%mktruns(help)
%mktruns(?)
```

%MktRuns Macro Notes

This macro specifies **options nonotes** throughout most of its execution. If you want to see all the notes, submit the following statement before running the macro:

```
%let mktopts = notes;
```

To see the macro version, submit the following statement before running the macro:

```
%let mktopts = version;
```

Example

When you invoke the macro, you usually specify one argument, a list of the number of levels of each factor. For example, with three 2-level and four 3-level factors, you can specify either of the following:

```
%mktruns(2 2 2 3 3 3 3)
%mktruns(2 ** 3 3 ** 4)
```

The output from the macro is shown in Figure 1.

Figure 1 %MktRuns Output

Design Summary	
Number of Levels	Frequency
2	3
3	4

Figure 1 *continued*

Saturated = 12
 Full Factorial = 648

Some Reasonable Design Sizes	Violations	Cannot Be Divided By
36 *	0	
72 *	0	
18	3	4
54	3	4
12 S	6	9
24	6	9
48	6	9
60	6	9
30	9	4 9
42	9	4 9

* - 100% Efficient design can be made with the MktEx macro.
 S - Saturated Design - The smallest design that can be made.

Figure 1 *continued*

n	Design	Reference
36	2**16 3**4	Orthogonal Array
36	2**11 3**12	Orthogonal Array
36	2**10 3**8 6**1	Orthogonal Array
36	2**9 3**4 6**2	Orthogonal Array
36	2**4 3**13	Orthogonal Array
36	2**3 3**9 6**1	Orthogonal Array
72	2**52 3**4	Orthogonal Array
72	2**49 3**4 4**1	Orthogonal Array
72	2**47 3**12	Orthogonal Array
72	2**46 3**8 6**1	Orthogonal Array
72	2**45 3**4 6**2	Orthogonal Array
72	2**44 3**12 4**1	Orthogonal Array
72	2**43 3**8 4**1 6**1	Orthogonal Array
72	2**42 3**4 4**1 6**2	Orthogonal Array
72	2**40 3**13	Orthogonal Array
72	2**39 3**9 6**1	Orthogonal Array
72	2**38 3**12 6**1	Orthogonal Array
72	2**38 3**5 6**2	Orthogonal Array
72	2**37 3**13 4**1	Orthogonal Array
72	2**37 3**8 6**2	Orthogonal Array
72	2**36 3**12 12**1	Orthogonal Array
72	2**36 3**9 4**1 6**1	Orthogonal Array
72	2**36 3**7 6**3	Orthogonal Array
72	2**35 3**12 4**1 6**1	Orthogonal Array
72	2**35 3**5 4**1 6**2	Orthogonal Array
72	2**34 3**8 4**1 6**2	Orthogonal Array
72	2**29 3**11 6**2	Orthogonal Array
72	2**27 3**11 6**1 12**1	Orthogonal Array
72	2**27 3**6 6**4	Orthogonal Array
72	2**23 3**24	Orthogonal Array
72	2**22 3**20 6**1	Orthogonal Array
72	2**21 3**16 6**2	Orthogonal Array
72	2**20 3**24 4**1	Orthogonal Array
72	2**20 3**12 6**3	Orthogonal Array
72	2**19 3**20 4**1 6**1	Orthogonal Array
72	2**19 3**8 6**4	Orthogonal Array
72	2**18 3**16 4**1 6**2	Orthogonal Array
72	2**18 3**7 6**5	Orthogonal Array
72	2**17 3**12 4**1 6**3	Orthogonal Array
72	2**16 3**25	Orthogonal Array
72	2**16 3**8 4**1 6**4	Orthogonal Array
72	2**15 3**21 6**1	Orthogonal Array
72	2**15 3**7 4**1 6**5	Orthogonal Array
72	2**14 3**24 6**1	Orthogonal Array
72	2**14 3**17 6**2	Orthogonal Array
72	2**13 3**25 4**1	Orthogonal Array
72	2**13 3**20 6**2	Orthogonal Array
72	2**13 3**13 6**3	Orthogonal Array

Figure 1 *continued*

n	Design	Reference
72	2**12 3**24 12**1	Orthogonal Array
72	2**12 3**21 4**1 6**1	Orthogonal Array
72	2**12 3**16 6**3	Orthogonal Array
72	2**12 3**9 6**4	Orthogonal Array
72	2**11 3**24 4**1 6**1	Orthogonal Array
72	2**11 3**20 6**1 12**1	Orthogonal Array
72	2**11 3**17 4**1 6**2	Orthogonal Array
72	2**11 3**12 6**4	Orthogonal Array
72	2**11 3**8 6**5	Orthogonal Array
72	2**10 3**20 4**1 6**2	Orthogonal Array
72	2**10 3**16 6**2 12**1	Orthogonal Array
72	2**10 3**13 4**1 6**3	Orthogonal Array
72	2**10 3**4 6**6	Orthogonal Array
72	2**9 3**16 4**1 6**3	Orthogonal Array
72	2**9 3**12 6**3 12**1	Orthogonal Array
72	2**9 3**9 4**1 6**4	Orthogonal Array
72	2**9 3**7 6**6	Orthogonal Array
72	2**8 3**12 4**1 6**4	Orthogonal Array
72	2**8 3**8 4**1 6**5	Orthogonal Array
72	2**8 3**8 6**4 12**1	Orthogonal Array
72	2**7 3**7 6**5 12**1	Orthogonal Array
72	2**7 3**4 4**1 6**6	Orthogonal Array
72	2**6 3**7 4**1 6**6	Orthogonal Array

The macro reports that the saturated design has 12 runs and that 36 and 72 are optimal design sizes. The macro picks 36, because it is the smallest integer greater than or equal to 12 that can be divided by 2, 3, 2×2 , 2×3 , and 3×3 . The macro also reports 18 as a reasonable size. There are three violations with 18, because 18 cannot be divided by each of the three pairs of 2×2 , so perfect orthogonality in the 2-level factors is not possible with 18 runs. Larger sizes are also reported. The macro displays orthogonal designs that are available from the %MktEx macro that match your specification.