

%MktRuns Macro

The `%MktRuns` autocall macro suggests reasonable sizes for experimental designs. See the following pages for examples of using this macro in the design chapter: 128, 188 and 199. Also see the following pages for examples of using this macro in the discrete choice chapter: 302, 340, 411, 415, 482, 483, 535 and 557. Additional examples appear throughout this chapter. The `%MktRuns` macro 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. Typically, the macro takes one argument, a list of the number of levels of each factor.

For example, with 3 two-level and 4 three-level factors, you can specify either of the following:

```
%mktruns(2 2 2 3 3 3)
```

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

The output from the macro is as follows:

Design Summary

Number of Levels	Frequency	
2	3	
3	4	
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.

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
.		
.		
.		

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 ≥ 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 two-level factors will not be possible with 18 runs. Larger sizes are reported as well. The macro displays orthogonal designs that are available from the `%MktEx` macro that match your specification.

You can run PROC PRINT after the macro finishes to see every size the macro considered, for example, as follows:

```
proc print label data=nums split='-' ;
  label s = '00'x;
  id n;
  run;
```

The output from this step is not shown.

For 2 two-level factors, 2 three-level factors, 2 four-level factors, and 2 five-level factors specify the following:

```
%mktruns(2 2 3 3 4 4 5 5)
```

The results are as follows:

Design Summary

Number of Levels	Frequency
2	2
3	2
4	2
5	2

Saturated = 21
 Full Factorial = 14,400

Some Reasonable Design Sizes	Violations	Cannot Be Divided By
120	3	9 16 25
180	6	8 16 25
60	7	8 9 16 25
144	15	5 10 15 20 25
48	16	5 9 10 15 20 25
72	16	5 10 15 16 20 25
80	16	3 6 9 12 15 25
96	16	5 9 10 15 20 25
160	16	3 6 9 12 15 25
192	16	5 9 10 15 20 25
21 S	34	2 4 5 6 8 9 10 12 15 16 20 25

S - Saturated Design - The smallest design that can be made.
 Note that the saturated design is not one of the recommended designs for this problem. It is shown to provide some context for the recommended sizes.

Among the smaller design sizes, 60 or 48 look like good possibilities.

The macro has an optional keyword parameter: `max=`. It specifies the maximum number of sizes to try. Usually you will not need to specify the `max=` option. The smallest design that is considered is the saturated design. The following specification tries 5000 sizes (21 to 5020) and reports that a perfect design can be found with 3600 runs:

```
%mktruns(2 2 3 3 4 4 5 5, max=5000)
```

The results are as follows:

Design Summary		
Number of Levels	Frequency	
2	2	
3	2	
4	2	
5	2	
Saturated	= 21	
Full Factorial	= 14,400	
Some Reasonable Design Sizes	Violations	Cannot Be Divided By
3600	0	
720	1	25
1200	1	9
1440	1	25
1800	1	16
2160	1	25
2400	1	9
2880	1	25
4320	1	25
4800	1	9
21 S	34	2 4 5 6 8 9 10 12 15 16 20 25

S - Saturated Design - The smallest design that can be made.
 Note that the saturated design is not one of the recommended designs for this problem. It is shown to provide some context for the recommended sizes.

The %MktEx macro does not explicitly know how to make this design, however, it can usually find it or come extremely close with the coordinate exchange algorithm.

Now consider again the problem with 3 two-level and 4 three-level factors, but this time we want to estimate the interaction of two of the two-level factors. The following step runs the macro:

```
%mktruns(2 2 2 3 3 3 3, interact=1*2, options=source)
```

Since options=source was specified, the first part of the following output lists the sources for orthogonality violations that the macro will consider. We see that n must be divided by: 2 since x_1 - x_3 are two-level factors, 3 since x_4 - x_7 are three-level factors, 4 since x_1*x_2 , x_1*x_3 , and x_2*x_3 interactions are specified, 6 since we have two-level and three-level factors, 8 since we have the two-way interaction of 2 two-level factors and the main-effect of an additional two-level factor, 9 since we have multiple three-

level factors, and 12 since we have the two-way interaction of 2 two-level factors and the main-effect of additional three-level factors. The results are as follows:

To Achieve Both Orthogonality and Balance, N Must Be Divided By		The Source of the Divisor	The Number of Sources of the Divisor	Failing to Divide by the Divisor Contributes this to the Violations	The Divisor Comes from these Factors
2		2	3	1 2 3	
3		3	4	4 5 6 7	
4		2*2	3	1 2 1 3 2 3	
6		2*3	12	1 4 1 5 1 6 1 7 2 4 2 5 2 6 2 7 3 4 3 5 3 6 3 7	
8		2*2*2	1	1 2 3	
9		3*3	6	4 5 4 6 4 7 5 6 5 7 6 7	

12	2*2*3	4	1 2 4
			1 2 5
			1 2 6
			1 2 7

Design Summary

Number of Levels	Frequency	
2	3	
3	4	
Saturated		= 13
Full Factorial		= 648
Some Reasonable Design Sizes	Violations	Cannot Be Divided By
72	0	
144	0	
36	1	8
108	1	8
48	6	9
96	6	9
120	6	9
60	7	8 9
84	7	8 9
13 S	33	2 3 4 6 8 9 12

S - Saturated Design - The smallest design that can be made.
 Note that the saturated design is not one of the recommended designs for this problem. It is shown to provide some context for the recommended sizes.

Now we need 72 runs for perfect balance and orthogonality although the %MktEx design catalog is not guaranteed to contain designs with interactions.

%MktRuns Macro Options

The following options can be used with the %MktRuns macro:

Option	Description
list	(positional) numbers of levels of all the factors
	(positional) “help” or “?” displays syntax summary
interact= <i>interaction-list</i>	interaction terms
max= <i>n</i> < <i>m</i> > < <i>f</i> >	largest design sizes to try

Option	Description
<code>n=n</code>	design size to evaluate
<code>maxlev=n</code>	maximum number of levels
<code>maxoa=n</code>	maximum number of orthogonal arrays
<code>options=justparse</code>	used by other <code>Mkt</code> macros to parse the list
<code>options=multiple</code>	allow terms to be counted multiple times
<code>options=multiple2</code>	<code>option=multiple</code> and more detailed output
<code>options=noprint</code>	suppress the display of all output
<code>options=nosat</code>	suppress the saturated design from the design list
<code>options=source</code>	displays source of numbers in design sizes
<code>options=512</code>	adds some designs in 512 runs
<code>out=SAS-data-set</code>	data set with the suggested sizes
<code>outorth=SAS-data-set</code>	data set with orthogonal array list
<code>toobig=n</code>	specifies problem that is too big

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

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

The `%MktRuns` macro has one positional parameter, `list`, and several keyword parameters.

list

specifies a list of the numbers of levels of all the factors. For example, for 3 two-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 4 three-level factors. You must specify a list. Note that the factor list is a positional parameter. This means it must come first, and unlike all other parameters, it is not specified after a name and an equal sign.

interact= *interaction-list*

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

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

The interaction syntax is in most ways like PROC GLM's and many of the other 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. Unlike PROC GLM's syntax, some short cuts are permitted. For the factor names, you can specify either the actual variable names (for example, `x1*x2` ...) or you can just specify 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 generating the orthogonal array list. The default is `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 $2 \leq n \leq 144$.

n= *n*

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

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. You can specify, for example, `maxoa=0, outorth=oa`, to get a data set instead of displaying this table. You could instead specify `maxoa=10` to just see the first ten arrays.

options= *options-list*

specifies binary options. By default, none of these options are specified. Specify one the following values after `options=`:

justparse

is used by other `Mkt` macros to have this macro just parse 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 counting orthogonality violations. 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 required as well, then the (1, 2) pair must have equal frequencies again. By default, each combination of variables is only counted once.

The difference between the single and multiple sources is the single source method just counts 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 for the two methods are often highly correlated, but they can be different.

multiple2

specifies both `option=multiple` and more detailed output including the reason each term appears 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 will only be added to the list if it meets the usual criteria for being a reasonable size.

source

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

512

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

out= *SAS-data-set*

specifies the name of a SAS data set with the suggested sizes. The default is `out=nums`.

outorth= *SAS-data-set*

specifies output data set with orthogonal array list. By default, this data set is not created.

toobig= *n*

is used to flag problems that are too big and take too long before they consume large quantities of resources. The default is `toobig=11400`. The number is the maximum number of levels and pairs of levels that is considered. With models with interactions, the interaction terms contribute to this number as well. A main-effects model with 150 factors is small enough ($150 \times 151/2 = 11325$) to work with the default `toobig=11400`. If you want to use the `%MktRuns` macro with larger models, you will have to specify a larger value for `toobig=`. The following step works with `toobig=` specified, but it is

too large to work without it being specified:

```
%mktruns(2 ** 17, interact=@2, toobig=20000)
```

%MktRuns Macro Notes

This macro specifies `options nonotes` throughout most of its execution. If you want to see all of the notes, submit the statement `%let mktopts = notes;` before running the macro. To see the macro version, submit the statement `%let mktopts = version;` before running the macro.