%MktBlock Macro

The `%MktBlock` autocall macro blocks a choice design or an ordinary linear experimental design. See the following pages for examples of using this macro in the discrete choice chapter: 426 and 497. Additional examples appear throughout this chapter. When a choice design is too large to show all choice sets to each subject, the design is blocked and a block of choice sets is shown to each subject. For example, if there are 36 choice sets, instead of showing each subject 36 sets, you could instead create 2 blocks and show 2 groups of subjects 18 sets each. You could also create 3 blocks of 12 choice sets or 4 blocks of 9 choice sets. You can also request just one block if you want to see the correlations and frequencies among all of the attributes of all of the alternatives of a choice design.

The design can be in one of two formats. Typically, a choice design has one row for each alternative of each choice set and one column for each of the attributes. Typically, this kind of design is produced by either the `%ChoicEff` or `%MktRoll` macro. Alternatively, a linear arrangement is an intermediate step in preparing some choice designs.* The linear arrangement has one row for each choice set and one column for each attribute of each alternative. Typically, the linear arrangement is produced by the `%MktEx` macro. The output from the `%MktBlock` macro is a data set containing the design, with the blocking variable added and hence not in the original order, with runs or choice sets nested within blocks.

The macro tries to create a blocking factor that is uncorrelated with every attribute of every alternative. In other words, the macro is trying to optimally add one additional factor, a blocking factor, to the linear arrangement. It is trying to make a factor that is orthogonal to all of the attributes of all of the alternatives. For linear arrangements, you can usually ask for a blocking factor directly as just another factor in the design, and then use the `%MktLab` macro to provide a name like `Block`, or you can use the `%MktBlock` macro.

The following steps create the blocking variable directly:

```
%mktex(3 ** 7, n=27, seed=350)

%mktlab(data=randomized, vars=x1-x6 Block)
```

The following steps create a design and then block it:

```
%mktex(3 ** 6, n=27, seed=350)

%mktblock(data=randomized, nblocks=3, seed=377, maxiter=50)
```

The results are as follows:

*See page 67 for an explanation of the linear arrangement of a choice design versus the arrangement of a choice design that is more suitable for analysis.

Canonical Correlations Between the Factors
There are 0 Canonical Correlations Greater Than 0.316

	Block	x1	x2	x3	x4	x5	x6
Block	1	0	0	0	0	0	0
x1	0	1	0	0	0	0	0
x2	0	0	1	0	0	0	0
x3	0	0	0	1	0	0	0
x4	0	0	0	0	1	0	0
x5	0	0	0	0	0	1	0
x6	0	0	0	0	0	0	1

Summary of Frequencies
There are 0 Canonical Correlations Greater Than 0.316

Frequencies

Block	9 9 9
x1	9 9 9
x2	9 9 9
x3	9 9 9
x4	9 9 9
x5	9 9 9
x6	9 9 9
Block x1	3 3 3 3 3 3 3 3 3
Block x2	3 3 3 3 3 3 3 3 3
Block x3	3 3 3 3 3 3 3 3 3
Block x4	3 3 3 3 3 3 3 3 3
Block x5	3 3 3 3 3 3 3 3 3
Block x6	3 3 3 3 3 3 3 3 3
x1 x2	3 3 3 3 3 3 3 3 3
x1 x3	3 3 3 3 3 3 3 3 3
x1 x4	3 3 3 3 3 3 3 3 3
x1 x5	3 3 3 3 3 3 3 3 3
x1 x6	3 3 3 3 3 3 3 3 3
x2 x3	3 3 3 3 3 3 3 3 3
x2 x4	3 3 3 3 3 3 3 3 3
x2 x5	3 3 3 3 3 3 3 3 3
x2 x6	3 3 3 3 3 3 3 3 3

```
x3 x4       3 3 3 3 3 3 3 3 3
x3 x5       3 3 3 3 3 3 3 3 3
x3 x6       3 3 3 3 3 3 3 3 3
x4 x5       3 3 3 3 3 3 3 3 3
x4 x6       3 3 3 3 3 3 3 3 3
x5 x6       3 3 3 3 3 3 3 3 3
N-Way       1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
            1 1 1 1 1 1 1 1 1
```

The output shows that the blocking factor is uncorrelated with all of the factors in the design. This output comes from the `%MktEval` macro, which is called by the `%MktBlock` macro.

The following output displays the blocked linear arrangement (3 blocks of nine choice sets):

Block	Run	x1	x2	x3	x4	x5	x6
1	1	2	1	2	2	1	1
	2	3	2	1	1	3	2
	3	1	3	3	3	2	3
	4	2	3	1	3	1	2
	5	1	1	1	2	2	2
	6	2	2	3	1	1	3
	7	3	1	3	2	3	3
	8	3	3	2	3	3	1
	9	1	2	2	1	2	1

Block	Run	x1	x2	x3	x4	x5	x6
2	1	1	3	3	1	3	2
	2	3	3	2	1	1	3
	3	2	2	3	2	2	2
	4	3	2	1	2	1	1
	5	2	1	2	3	2	3
	6	3	1	3	3	1	2
	7	2	3	1	1	2	1
	8	1	2	2	2	3	3
	9	1	1	1	3	3	1

Block	Run	x1	x2	x3	x4	x5	x6
3	1	3	2	1	3	2	3
	2	3	1	3	1	2	1
	3	1	3	3	2	1	1
	4	2	2	3	3	3	1
	5	2	1	2	1	3	2
	6	1	2	2	3	1	2
	7	3	3	2	2	2	2
	8	2	3	1	2	3	3
	9	1	1	1	1	1	3

Note that in the linear version of the design, there is one row for each choice set and all of the attributes of all of the alternatives are in the same row.

Next, we will create and block a choice design with two blocks of nine sets instead of blocking the linear version of a choice design. The following steps create and then block a choice design:

```
%mktex(3 ** 6, n=3**6)

%mktroll(design=design, key=2 3, out=out)

%choiceff(data=out,                  /* candidate set of choice sets         */
          model=class(x1-x3 / sta), /* model with stdz orthogonal coding    */
          nsets=18,                  /* number of choice sets                */
          nalts=2,                   /* number of alternatives               */
          seed=151,                  /* random number seed                   */
          options=nodups             /* do not create duplicate choice sets  */
                  relative,          /* display relative D-efficiency        */
          beta=zero)                 /* assumed beta vector, Ho: b=0         */

* Block the choice design.  Ask for 2 blocks;
%mktblock(data=best, nalts=2, nblocks=2, factors=x1-x3, seed=472)
```

(Note that if this had been a branded example, and if you are running SAS version 8.2 or an earlier release, specify `id=brand`; do not add your brand variable to the factor list. For SAS 9.0 and later SAS releases, it is fine to add your brand variable to the factor list.)

Both the design and the blocking are not as good this time. The variable names in the output are composed of `Alt`, the alternative number, and the factor name. Since there are two alternatives each composed of three factors plus one blocking variable ($2 \times 3 + 1 = 7$), a 7×7 correlation matrix is reported. Some of the results are as follows:

Canonical Correlations Between the Factors
There are 7 Canonical Correlations Greater Than 0.316

	Block	Alt1_x1	Alt1_x2	Alt1_x3	Alt2_x1	Alt2_x2	Alt2_x3
Block	1	0.15	0	0.14	0.13	0.15	0.14
Alt1_x1	0.15	1	0.41	0.21	0.51	0.20	0.30
Alt1_x2	0	0.41	1	0.40	0.26	0.56	0.33
Alt1_x3	0.14	0.21	0.40	1	0.19	0.31	0.52
Alt2_x1	0.13	0.51	0.26	0.19	1	0.31	0.30
Alt2_x2	0.15	0.20	0.56	0.31	0.31	1	0.48
Alt2_x3	0.14	0.30	0.33	0.52	0.30	0.48	1

Summary of Frequencies
There are 7 Canonical Correlations Greater Than 0.316
* - Indicates Unequal Frequencies

		Frequencies
	Block	9 9
*	Alt1_x1	8 5 5
*	Alt1_x2	4 6 8
*	Alt1_x3	6 7 5
*	Alt2_x1	4 7 7
*	Alt2_x2	8 5 5
*	Alt2_x3	6 5 7
*	Block Alt1_x1	4 2 3 4 3 2
*	Block Alt1_x2	2 3 4 2 3 4
*	Block Alt1_x3	3 3 3 3 4 2
*	Block Alt2_x1	2 4 3 2 3 4
*	Block Alt2_x2	4 3 2 4 2 3
*	Block Alt2_x3	3 3 3 3 2 4
*	Alt1_x1 Alt1_x2	3 3 2 1 1 3 0 2 3
*	Alt1_x1 Alt1_x3	3 3 2 2 2 1 1 2 2
*	Alt1_x1 Alt2_x1	0 4 4 2 0 3 2 3 0
*	Alt1_x1 Alt2_x2	3 3 2 2 1 2 3 1 1
*	Alt1_x1 Alt2_x3	3 3 2 2 1 2 1 1 3
*	Alt1_x2 Alt1_x3	2 2 0 3 1 2 1 4 3
*	Alt1_x2 Alt2_x1	1 2 1 1 3 2 2 2 4
*	Alt1_x2 Alt2_x2	0 2 2 3 0 3 5 3 0
*	Alt1_x2 Alt2_x3	1 1 2 3 2 1 2 2 4
*	Alt1_x3 Alt2_x1	1 3 2 2 2 3 1 2 2
*	Alt1_x3 Alt2_x2	3 2 1 3 1 3 2 2 1
*	Alt1_x3 Alt2_x3	0 3 3 3 0 4 3 2 0

```
*    Alt2_x1 Alt2_x2    1 1 2 3 3 1 4 1 2
*    Alt2_x1 Alt2_x3    1 2 1 3 1 3 2 2 3
*    Alt2_x2 Alt2_x3    1 2 5 2 2 1 3 1 1
     N-Way              1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

Note that in this example, the input is a choice design (as opposed to the linear version of a choice design) so the results are in choice design format. There is one row for each alternative of each choice set. Some of the results are as follows:

Block	Set	Alt	x1	x2	x3
1	1	1	3	2	3
		2	2	3	1
1	2	1	2	1	2
		2	1	3	1
.					
.					
.					
Block	Set	Alt	x1	x2	x3
2	1	1	2	2	1
		2	1	3	2
2	2	1	2	3	1
		2	3	1	3
.					
.					
.					

%MktBlock Macro Options

The following options can be used with the `%MktBlock` macro:

Option	Description
`help`	(positional) "help" or "?" displays syntax summary
`alt=`*variable*	alternative number variable
`block=`*variable*	block number variable
`data=`*SAS-data-set*	either the choice design or linear arrangement
`factors=`*variable-list*	factors in the design
`id=`*variable-list*	variables to copy to output data set
`initblock=`*variable*	initial blocking variable

Option	Description
`iter=`*n*	times to try to block the design
`list=`*n*	list larger canonical correlations
`maxiter=`*n*	times to try to block the design
`nalts=`*n*	number of alternatives in choice set
`nblocks=`*n*	number of blocks to create
`next=`*n*	where to look for the next exchange
`options=nosort`	do not sort the design into blocks
`out=`*SAS-data-set*	output data set with block numbers
`outr=`*SAS-data-set*	randomized output data set
`print=`*print-options*	output display options
`ridge=`*n*	ridging factor
`seed=`*n*	random number seed
`set=`*variable*	choice set number variable
`vars=`*variable-list*	factors in the design

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

```
%mktblock(help)
%mktblock(?)
```

alt= *variable*

specifies the alternative number variable. If this variable is in the input data set, it is excluded from the factor list. The default is `alt=Alt`.

block= *variable*

specifies the block number variable. If this variable is in the input data set, it is excluded from the factor list. The default is `block=Block`.

data= *SAS-data-set*

specifies either the choice design or the linear arrangement. The choice design has one row for each alternative of each choice set and one column for each of the attributes. Typically, this design is produced by either the `%ChoicEff` or `%MktRoll` macro. For choice designs, you must also specify the `nalts=` option. By default, the macro uses the last data set created. The linear arrangement has one row for each choice set and one column for each attribute of each alternative. Typically, this design is produced by the `%MktEx` macro. This is the design that is input into the `%MktRoll` macro.

factors= *variable-list*
vars= *variable-list*

specifies the factors in the design. By default, all numeric variables are used, except variables with names matching those specified in the `block=`, `set=`, `alt=`, and `id=` options. (By default, the variables `Block`, `Set`, `Run`, and `Alt` are excluded from the factor list.) If you are using version 8.2 or an earlier SAS release with a branded choice design (assuming the brand factor is called `Brand`), specify `id=Brand`. Do not add the brand factor to the factor list unless you are using SAS 9.0 or a later SAS release.

id= *variable-list*
specifies the `data=` data set variables to copy to the output data set. If you are using version 8.2 or an earlier SAS release with a branded choice design (assuming the brand factor is called `Brand`), specify `id=Brand`. Do not add the brand factor to the factor list unless you are using SAS 9.0 or a later SAS release.

initblock= *variable*
specifies the name of the variable in the data set that is to be used as the initial blocking variable for the first iteration.

list= r
lists canonical correlations larger than `list=`r. The default is $r = 0.316 \approx \sqrt{r^2 = 0.1}$.

maxiter= n
iter= n
specifies the number of times to try to block the design starting with a different random blocking. By default, the macro tries five random starts, and iteratively refines each until D-efficiency quits improving, then in the end selects the blocking with the best D-efficiency.

nalts= n
specifies the number of alternatives in each choice set. If you are inputting a choice design, you must specify `nalts=`, otherwise the macro assumes you are inputting a linear arrangement.

nblocks= n
specifies the number of blocks to create. The option `nblocks=1` just reports information about the design. The `nblocks=` option must be specified.

next= n
specifies how far into the design to go to look for the next exchange. The specification `next=1` specifies that the macro should try exchanging the level for each run with the level for the next run and all other runs. The specification `next=2` considers exchanges with half of the other runs, which makes the algorithm run more quickly. The macro considers exchanging the level for run i with run $i + 1$ then uses the `next=` value to find the next potential exchanges. Other values, including nonintegers can be specified as well. For example `next=1.5` considers exchanging observation 1 with observations 2, 4, 5, 7, 8, 10, 11, and so on. With smaller values, the macro tends to find a slightly better blocking variable at a cost of much slower run time.

options= *options-list*
specifies binary options. By default, no binary options are specified. Specify the following value after `options=`.

`nosort`
do not sort the design into blocks. This is useful anytime you want the order of the observations in the output data set to match the order of the observations in the input data set. You will typically not want to specify `options=nosort` when you are using the `%MktBlock` macro to block a design. However, `options=nosort` is handy when you are using the `%MktBlock` macro to add just another factor to the design.

out= *SAS-data-set*
specifies the output data set with the block numbers. The default is `out=blocked`. Often, you will want to specify a two-level name to create a permanent SAS data set so the design is available later for analysis.

outr= *SAS-data-set*
specifies the randomized output data set if you would like the design randomly sorted within blocks. Often, you will want to specify a two-level name to create a permanent SAS data set so the design is available later for analysis.

print= *print-options*
specifies both the `%MktBlock` and the `%MktEval` macro display options, which control the display of the results. The default is `print=normal`. Specify one or more values from the following list.

`all`	all output is displayed
`corr`	canonical correlations
`block`	canonical correlations within blocks
`design`	blocked design
`freqs`	long frequencies list
`list`	list of big canonical correlations
`nonzero`	like `ordered` but sets `list=1e-6`
`noprint`	no output is displayed
`normal`	`corr list summ design note`
`note`	blocking note
`ordered`	like `list` but ordered by variable names
`short`	`corr summ note`
`summ`	frequency summaries

ridge= *n*
specifies the value to add to the diagonal of $\mathbf{X}'\mathbf{X}$ to make it nonsingular. Usually, you will not need to change this value. If you do, you probably will not notice any effect. Specify `ridge=0` to use a generalized inverse instead of ridging. The default is `ridge=0.01`.

seed= *n*
specifies the random number seed. By default, `seed=0`, and clock time is used to make the random number seed. By specifying a random number seed, results should be reproducible within a SAS release for a particular operating system and for a particular version of the macro. However, due to machine and macro differences, some results might not be exactly reproducible everywhere, although you would expect the efficiency differences to be slight.

set= *variable*
specifies the choice set number variable. When `nalts=` is specified, the default is `Set`, otherwise the default is `Run`. If this variable is in the input data set, it is excluded from the factor list.

%MktBlock 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.