

The %MktRoll Macro

Introduction

The %MktRoll autocall macro constructs a choice design from a linear arrangement. It takes as input a SAS data set that contains an experimental design that has one row per choice set (the linear arrangement). This data set is specified in the macro's DESIGN= argument. The DESIGN= data set has one variable for each attribute of each alternative in the choice experiment. The output from the %MktRoll macro is an OUT= SAS data set. The OUT= data set is the choice design that contains the experimental design that has one row per alternative per choice set. There is one column for each different attribute. For example, in a simple branded study, the DESIGN= data set might contain the variables X1–X5, which contain the prices of each of five alternative brands. The OUT= data set would have one factor, Price, that contains the price of each of the five alternatives. In addition, it would have the number (or optionally the name) of each alternative.

The rules for determining the mapping between factors in the DESIGN= data set and the OUT= data set are contained in the KEY= data set. For example, assume that the DESIGN= data set contains the variables X1–X5, which contain the prices of each of five alternative brands: Brand A, B, C, D, and E. The choice design has two factors, Brand and Price. Brand A price is made from X1, Brand B price is made from X2, Brand C price is made from X3, and so on. A convenient way to get all the names in a variable list like X1–X5 is to use the %MktKey macro. The following step creates the five names in a single column:

```
%mktkey(5 1)
```

The %MktKey macro produces the following data set:

x1

x1

x2

x3

x4

x5

The following step creates the Key data set:

```
data key;  
  input (Brand Price) ($);  
  datalines;  
A x1  
B x2  
C x3  
D x4  
E x5  
;
```

This data set has two variables. Brand contains the brand names, and Price contains the names of the factors that are used to make the price effects for each of the alternatives. The OUT= data set will contain the variables that have the same names as the variables in the KEY= data set.

The following step creates the linear arrangement, with one row per choice set:

```
%mktex(3 ** 5, n=12)
```

The following step creates the choice design, with one row per alternative per choice set:

```
%mktroll(design=randomized, key=key, OUT=sasuser.design, alt=brand)
```

For example, consider a randomized data set that contains the following row:

Obs	x1	x2	x3	x4	x5
9	3	1	1	2	1

Then the data set Sasuser.Design contains the following rows:

Obs	Set	Brand	Price
41	9	A	3
42	9	B	1
43	9	C	1
44	9	D	2
45	9	E	1

The price for Brand A is made from X1=3, the price for Brand B is made from X2=1, and so on.

Now assume that there are three alternatives, each a different brand, and each composed of four factors: Price, Size, Color, and Shape. In addition, there is a constant alternative. First, the %MktEx macro is used to create a design that has 12 factors, one for each attribute of each alternative. The following step creates the design:

```
%mktex(2 ** 12, n=16, seed=109)
```

The following DATA step creates the KEY= data set:

```
data key;
  input (Brand Price Size Color Shape) ($);
  datalines;
A      x1      x2      x3      x4
B      x5      x6      x7      x8
C      x9      x10     x11     x12
None   .       .       .       .
;
```

It shows that there are three brands, A, B, and C, and also None.

Brand A is created from Brand = “A,” Price = X1, Size = X2, Color = X3, and Shape = X4.

Brand B is created from Brand = “B,” Price = X5, Size = X6, Color = X7, and Shape = X8.

Brand C is created from Brand = “C,” Price = X9, Size = X10, Color = X11, and Shape = X12.

The constant alternative is created from Brand = “None” and none of the attributes. The “.” notation indicates missing values in input data sets. The actual values in the Key data set are blank (missing character).

The following step creates the design, with one row per alternative per choice set:

```
%mktroll(design=randomized, key=key, out=sasuser.design, alt=brand)
```

For example, consider a randomized data set that contains the following row:

Obs	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12
8	2	2	2	2	2	1	1	2	2	2	1	2

Then the data set Sasuser.Design contains the following rows:

29	8	A	2	2	2	2
30	8	B	2	1	1	2
31	8	C	2	2	1	2
32	8	None

Now assume like before that there are three branded alternatives, each composed of four factors: Price, Size, Color, and Shape. In addition, there is a constant alternative. Also, there is an alternative-specific factor, Pattern, that only applies to Brand A and Brand C. First, the %MktEx macro is used to create a design that has 14 factors, one for each attribute of each alternative. The following step creates the design:

```
%mktex(2 ** 14, n=16, seed=114)
```

The following DATA step creates the KEY= data set:

```
data key;
  input (Brand Price Size Color Shape Pattern) ($);
  datalines;
A      x1      x2      x3      x4      x13
B      x5      x6      x7      x8      .
C      x9      x10     x11     x12     x14
None   .       .       .       .       .
;
```

It shows that there are three brands, A, B, and C, plus None.

Brand A is created from Brand = “A,” Price = X1, Size = X2, Color = X3, Shape = X4, and Pattern = X13.

Brand B is created from Brand = “B,” Price = X5, Size = X6, Color = X7, and Shape = X8.

Brand C is created from Brand = “C,” Price = X9, Size = X10, Color = X11, Shape = X12, and Pattern = X14.

The constant alternative is Brand = “None” and none of the attributes.

The following step creates the design, with one row per alternative per choice set:

```
%mktroll(design=randomized, key=key, out=sasuser.design, alt=brand)
```

For example, consider a randomized data set that contains the following row:

Obs	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14
8	2	1	1	2	1	2	1	2	1	1	1	2	1	2

Then the data set Sasuser.Design contains the following rows:

Obs	Set	Brand	Price	Size	Color	Shape	Pattern
29	8	A	2	1	1	2	1
30	8	B	1	2	1	2	.
31	8	C	1	1	1	2	2
32	8	None

Now suppose you are going to fit a model with price cross-effects; you need X1, X5, and X9 (the three price effects) available in the OUT= data set. The following step creates the design:

```
%mktroll(design=randomized, key=key, out=sasuser.design,
          alt=brand, keep=x1 x5 x9)
```

Now the data set also contains the three original price variables, for example, as follows:

Obs	Set	Brand	Price	Size	Color	Shape	Pattern	x1	x5	x9
29	8	A	2	1	1	2	1	2	1	1
30	8	B	1	2	1	2	.	2	1	1
31	8	C	1	1	1	2	2	2	1	1
32	8	None	2	1	1

Every value in the KEY= data set must appear as a variable in the DESIGN= data set. The %MktRoll macro displays a warning if it encounters a variable name in the DESIGN= data set that does not appear as a value in the KEY= data set.

%MktRoll Macro Syntax

```
%MktRoll(DESIGN=SAS-data-set, KEY=SAS-data-set <, optional arguments >)
```

Required Arguments

DESIGN=SAS-data-set

specifies an input SAS data set that has one row per choice set.

KEY=SAS-data-set | list

specifies the rules for mapping the DESIGN= data set to the OUT= data set. It has one of the following two forms:

- The KEY= option names an input SAS data set that contains the rules for mapping the DESIGN= data set to the OUT= data set. The preceding examples describe the structure of this data set in detail.
- When you want the KEY= data set to exactly match the data set that comes out of the %MktKey macro, you can specify the argument to the %MktKey macro directly in the KEY= option, and the %MktRoll macro will make the KEY=Key data set for you. In other words, the following two specifications are equivalent:

```
%mktkey(3 3 t)
%mktroll(design=design, key=key, out=frac)

%mktroll(design=design, key=3 3 t, out=frac)
```

Optional Arguments

ALT=variable

specifies the variable in the KEY= data set that contains the name of each alternative. Often this is something like ALT=Brand. When you do not specify ALT=, the macro creates a variable named _Alt_ that contains the alternative number.

KEEP=variable-list

specifies factors from the DESIGN= data set that should also be kept in the OUT= data set. This option is useful for keeping terms that are used to create cross-effects.

OPTIONS=options-list

specifies binary options. You can specify one or more of the following:

NOWARN

does not display a warning when the DESIGN= data set contains variables that do not appear in the KEY= data set.

OUT=SAS-data-set

specifies the output SAS data set. If you do not specify OUT=, the DATA*n* convention is used.

SET=variable

specifies the variable in the OUT= data set that will contain the choice set number. By default, this variable is named Set.

Help Option

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

```
%mktroll(help)
%mktroll(?)
```

%MktRoll 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;
```