

# ChoicEff Macro Options

Option	Description
<code>bestcov=SAS-data-set</code>	covariance matrix for the best design
<code>bestout=SAS-data-set</code>	best design
<code>beta=list</code>	true parameters
<code>chunks=n</code>	number of observations to code at once
<code>converge=n</code>	convergence criterion
<code>cov=SAS-data-set</code>	all of the covariance matrices
<code>data=SAS-data-set</code>	input choice candidate set
<code>drop=variable-list</code>	variables to drop from the model
<code>fixed=variable-list</code>	variable that flags fixed alternatives
<code>flags=variable-list</code>	variables that flag the alternatives
<code>init=SAS-data-set</code>	input initial design data set
<code>initvars=variable-list</code>	initial variables
<code>intiter=n</code>	maximum number of internal iterations
<code>iter=n</code>	maximum iterations (designs to create)
<code>maxiter=n</code>	maximum iterations (designs to create)
<code>model=model-specification</code>	<code>model</code> statement list of effects
<code>morevars=variable-list</code>	more variables to add to the model
<code>n=n</code>	number of observations
<code>nalts=n</code>	number of alternatives
<code>nsets=n</code>	number of choice sets desired
<code>options=coded</code>	displays the coded candidate set
<code>options=detail</code>	displays the details of the swaps
<code>options=nocode</code>	skips the PROC TRANSREG coding stage
<code>options=nodups</code>	prevents duplicate choice set creation
<code>options=notests</code>	suppress the hypothesis tests
<code>options=orthcan</code>	orthogonalizes the candidate set
<code>options=relative</code>	displays final relative $D$ -efficiency
<code>out=SAS-data-set</code>	all designs data set
<code>rscale=n</code>	relative efficiency scaling factor
<code>seed=n</code>	random number seed
<code>submat=number-list</code>	submatrix for efficiency calculations
<code>types=integer-list</code>	number of sets of each type
<code>typevar=variable</code>	choice set types variable
<code>weight=weight-variable</code>	optional weight variable

```
%mktex(3 ** 3, n = 3**3) /* candidate set of alternatives */
```

```
%mktlab(int=f1-f3)
```

```
%choicEff(data=final, model=class(x1-x3 / sta), nsets=9, options=relative,
maxiter=10, seed=104, flags=f1-f3, beta=zero)
```

```
%mktex(3 ** 9, n=2187) /* candidate set of choice sets */

%mkkey(3 3)

%mkroll(design=randomized, key=key, out=frac)

%choicetf(data=frac, model=class(x1-x3 / sta), seed=104, maxiter=10,
           options=relative, nsets=9, nalts=3, beta=zero)

%mktx(3 ** 4, n = 9) /* design evaluation */

%mklab(data=design, vars=Set x1-x3)

%choicetf(data=final, model=class(x1-x3 / sta), nsets=3, nalts=3,
           options=relative, beta=zero, init=final(keep=set), intiter=0)
```

# MktAllo Macro Options

Option	Description
<code>data=SAS-data-set</code>	input SAS data set
<code>freq=variable</code>	frequency variable
<code>nalts=n</code>	number of alternatives
<code>out=SAS-data-set</code>	output SAS data set
<code>vars=variable-list</code>	input variables

```

data allocs2;
  input Brand $ 1-8 Price $ Count Set;
  datalines;
.          .          0 1
Brand 1    $50        103 1
Brand 2    $75         58 1
Brand 3    $50        318 1
Brand 4    $100        99 1
Brand 5    $100        54 1
Brand 6    $100        83 1
Brand 7    $75         71 1
Brand 8    $75         58 1
Brand 9    $75        100 1
Brand 10   $50         56 1
.          .          10 2
Brand 1    $100        73 2
Brand 2    $100        76 2
Brand 3    $100       342 2
Brand 4    $50         55 2
Brand 5    $75         50 2
Brand 6    $100        77 2
Brand 7    $75         95 2
Brand 8    $100        71 2
Brand 9    $50         72 2
Brand 10   $100        79 2
;

%mktallo(data=allocs2, out=allocs3, nalts=11,
  vars=set brand price, freq=Count)

```

# MktBal Macro Options

Option	Description
<code>list</code>	(positional) list of the numbers of levels
<code>init=SAS-data-set</code>	initial input experimental design
<code>iter=n</code>	maximum iterations (designs to create)
<code>maxinititer=n</code>	maximum initialization iterations
<code>maxiter=n</code>	maximum iterations (designs to create)
<code>maxstarts=n</code>	maximum number of random starts
<code>maxtries=n</code>	times to try refining each factor
<code>n=n</code>	number of runs in the design
<code>options=noprint</code>	suppress the final <i>D</i> -efficiency
<code>options=nosort</code>	do not sort the final design
<code>options=oa</code>	an orthogonal array is sought
<code>options=progress</code>	reports on macro progress
<code>options=sequential</code>	factors are added but not refined
<code>out=SAS-data set</code>	output experimental design
<code>seed=n</code>	random number seed

```
%mktbal(2 2 3 3 3, n=18, seed=151)
```

```
%mktbal(2 ** 23, n=24, options=oa progress, maxstarts=2000, seed=104)
```

```
%mktex(10 ** 4, n=100)
```

```
%mktbal(10 ** 4 2 2, n=100, init=design, options=oa progress,  
maxstarts=5000, seed=104, out=balanced)
```

# MktBIBD Macro Options

Option	Description
<code>b=b</code>	number of blocks (alias for <code>nsets=</code> )
<code>group=n</code>	number of groups
<code>k=k</code>	block size (alias for <code>setsize=</code> )
<code>nattrs=t</code>	number of attributes (alias for <code>t=</code> )
<code>nsets=b</code>	number of sets (alias for <code>b=</code> )
<code>optiter=n1 &lt; n2 &lt; n3 &gt;&gt;</code>	number of PROC OPTEX iterations
<code>options=position</code>	optimizes position frequencies
<code>options=neighbor</code>	nondirectional row-neighbor balance and position
<code>options=serial</code>	directional row-neighbor balance and position
<code>out=SAS-data-set</code>	output data set BIBD
<code>outf=SAS-data-set</code>	output data set factorial design
<code>outi=SAS-data-set</code>	output data set incidence matrix
<code>outs=SAS-data-set</code>	output data set sorted design
<code>positer=n1 &lt; n2 &lt; n3 &gt;&gt;</code>	number of iterations position frequencies
<code>seed=n</code>	random number seed
<code>setsize=k</code>	set size (alias for <code>k=</code> )
<code>t=t</code>	number of treatments (alias for <code>nattrs=</code> )
<code>weights=n n</code>	position frequency badness weights

```
%mktbibd(b=10, t=5, k=3, seed=104)
```

```
%mktbibd(nsets=10, nattrs=5, setsize=3, seed=104)
```

```
%mktbibd(b=14, t=7, k=4, options=neighbor, seed=104)
```

```
%mktbibd(b=14, t=7, k=4, options=serial, seed=361699)
```

# MktBlock Macro Options

Option	Description
<code>alt=variable</code>	alternative number variable
<code>block=variable</code>	block number variable
<code>data=SAS-data-set</code>	either the choice or linear design
<code>factors=variable-list</code>	factors in the design
<code>id=variable-list</code>	variables to copy to output data set
<code>initblock=variable</code>	initial blocking variable
<code>iter=n</code>	times to try to block the design
<code>list=n</code>	list larger canonical correlations
<code>maxiter=n</code>	times to try to block the design
<code>nalts=n</code>	number of alternatives in choice set
<code>nblocks=n</code>	number of blocks to create
<code>next=n</code>	where to look for the next exchange
<code>options=nosort</code>	do not sort the design into blocks
<code>out=SAS-data-set</code>	output data set with block numbers
<code>outr=SAS-data-set</code>	randomized output data set
<code>print=print-options</code>	output display options
<code>ridge=n</code>	ridging factor
<code>seed=n</code>	random number seed
<code>set=variable</code>	choice set number variable
<code>vars=variable-list</code>	factors in the design

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

```
%mktblock(data=randomized, nblocks=3, factors=x1-x6, seed=377)
```

```
%mktex(3 ** 6, n=729)
```

```
%mktkey(2 3)
```

```
%mktroll(design=randomized, key=key, out=out)
```

```
%choicelf(data=out, model=class(x1-x3), nsets=18, nalts=2,  
beta=zero, options=nodups, seed=151)
```

```
%mktblock(data=best, nalts=2, nblocks=2, factors=x1-x3, seed=472)
```

# MktBSize Macro Options

Option	Description
<code>b=do-list</code>	number of blocks (alias for <code>nsets=</code> )
<code>k=do-list</code>	block size (alias for <code>setsize=</code> )
<code>maxreps=n</code>	maximum number of replications
<code>nattrs=do-list</code>	number of attributes (alias for <code>t=</code> )
<code>nsets=do-list</code>	number of sets (alias for <code>b=</code> )
<code>order=order-list</code>	order of the loops
<code>options=nocheck</code>	suppress checking <i>b</i> , <i>t</i> , and <i>k</i>
<code>options=ubd</code>	lifts the balance restriction on the design
<code>out=SAS-data-set</code>	output data set design list
<code>setsize=do-list</code>	set size (alias for <code>k=</code> )
<code>t=do-list</code>	number of treatments (alias for <code>nattrs=</code> )

```
%mktbsize(t=12, k=4 to 8, b=12 to 30)
```

```
%mktbsize(nattrs=12, setsize=4 to 8, nsets=12 to 30)
```

```
%mktbsize(t=5 to 20, k=3 to t - 1, nsets=t to 30)
```

```
%mktbsize(nattrs=5 to 20, setsize=3 to t - 1, nsets=t to 30)
```

```
%mktbsize(t=20, k=6, options=ubd)
```

```
%mktbsize(nattrs=20, setsize=6, options=ubd)
```

# MktDes Macro Options

Option	Description
<code>big=<i>n</i></code>	size of big candidate set
<code>cand=<i>SAS-data-set</i></code>	candidate design
<code>classopts=<i>options</i></code>	<b>class</b> statement options
<code>coding=<i>name</i></code>	<b>coding=</b> option
<code>examine=&lt; I &gt; &lt; V &gt;</code>	matrices that you want to examine
<code>facopts=<i>options</i></code>	PROC FACTEX statement options
<code>factors=<i>factor-list</i></code>	factors and levels for each factor
<code>generate=<i>options</i></code>	<b>generate</b> statement options
<code>interact=<i>interaction-list</i></code>	interaction terms
<code>iter=<i>n</i></code>	number of designs
<code>keep=<i>n</i></code>	number of designs to keep
<code>maxiter=<i>n</i></code>	number of designs
<code>method=<i>name</i></code>	search method
<code>n=<i>n</i>   SATURATED</code>	number of runs
<code>nlev=<i>n</i></code>	number of levels for pseudo-factors
<code>options=allcode</code>	shows all code
<code>options=check</code>	checks the <code>cand=</code> design
<code>options=nocode</code>	suppress the procedure code display
<code>otherfac=<i>variable-list</i></code>	other factors
<code>otherint=<i>terms</i></code>	multi-step interaction terms
<code>out=<i>SAS-data-set</i></code>	output experimental design
<code>procopts=<i>options</i></code>	PROC OPTEX statement options
<code>run=<i>procedure-list</i></code>	list of procedures that can be run
<code>seed=<i>n</i></code>	random number seed
<code>size=<i>n</i>   MIN</code>	candidate-set size
<code>step=<i>n</i></code>	step number
<code>where=<i>where-clause</i></code>	<b>where</b> clause

```
%mktdes(factors=x1-x2=2 x3-x5=3, n=18)
```

```
%mktdes(factors=x1-x3=2 x4-x6=3 x7-x9=5, n=30)
```

```
%mktdes(factors=x1-x3=2, n=30, run=factex, step=1)
```

```
%mktdes(factors=x4-x6=3, n=30, run=factex, step=2)
```

```
%mktdes(factors=x7-x9=5, n=30, run=factex optex, step=3)
```

```
%mktdes(factors=x1-x3=2 x4-x6=3(3) x7-x9=4, n=32, size=1024,
interact=x1|x2|x3@2 x4*x7)
```

# MktDups Macro Options

Option	Description
options	(positional) binary options
data= <i>SAS-data-set</i>	input choice design
factors= <i>variable-list</i>	factors in the design
nalts= <i>n</i>	number of alternatives
out= <i>SAS-data-set</i>	output data set
outlist= <i>SAS-data-set</i>	output data set with duplicates
vars= <i>variable-list</i>	factors in the design

```
%mktex(3 ** 3 2 ** 2, n=19, seed=513)

%mktdups(linear, factors=x1-x5);

%mktx(2 ** 5, n=2**5)

%mktlab(int=f1-f4, values=1 2)

%choicetex(data=final, model=class(x1-x5), seed=109,
            nsets=42, flags=f1-f4, beta=zero);

%mktdups(generic, data=best, factors=x1-x5, nalts=4, out=out)

%mktx(3 ** 9, n=27)

data key;
  input (Brand x1-x3) ($) @@;
  datalines;
Acme x1 x2 x3 Ajax x4 x5 x6 Widgit x7 x8 x9
;
%mktroll(design=randomized, key=key, alt=brand, out=cand);
%choicetex(data=cand, model=class(brand x1-x3), seed=420,
            nsets=18, nalts=3, beta=zero);
%mktdups(branded, data=best, factors=brand x1-x3, nalts=3, out=out)
```

# MktEval Macro Options

Option	Description
<code>blocks=variable</code>	blocking variable
<code>data=SAS-data-set</code>	input data set with design
<code>factors=variable-list</code>	factors in the design
<code>format=format</code>	format for canonical correlations
<code>freqs=frequency-list</code>	frequencies to display
<code>list=n</code>	minimum canonical correlation to list
<code>outcb=SAS-data-set</code>	within-block canonical correlations
<code>outcorr=SAS-data-set</code>	canonical correlation matrix
<code>outfreq=SAS-data-set</code>	frequencies
<code>outfsum=SAS-data-set</code>	frequency summaries
<code>outlist=SAS-data-set</code>	list of largest canonical correlations
<code>print=print-options</code>	controls the display of the results
<code>vars=variable-list</code>	factors in the design

```
%mktex(2 ** 2 3 ** 3, n=18, seed=205)
```

```
%mkteval;
```

# MktEx Macro Options

Option	Description
<code>list</code>	(positional) list of the numbers of factor levels
<code>anneal=<i>n1</i> &lt; <i>n2</i> &lt; <i>n3</i> &gt;&gt;</code>	starting probability for annealing
<code>annealfun=<i>function</i></code>	annealing probability function
<code>anniter=<i>n1</i> &lt; <i>n2</i> &lt; <i>n3</i> &gt;&gt;</code>	first annealing iteration
<code>balance=<i>n</i></code>	maximum level-frequency range
<code>big=<i>n</i> &lt; <i>choose</i> &gt;</code>	size of big full-factorial design
<code>canditer=<i>n1</i> &lt; <i>n2</i> &gt;</code>	iterations for OPTEx designs
<code>cat=<i>SAS-data-set</i></code>	input design catalog
<code>detfuzz=<i>n</i></code>	determinants change increment
<code>examine=&lt; I &gt; &lt; V &gt; &lt; <i>aliasing</i> &gt;</code>	matrices that you want to examine
<code>exchange=<i>n</i></code>	number of factors to exchange
<code>fixed=<i>variable</i></code>	indicates runs that are fixed
<code>holdouts=<i>n</i></code>	adds holdout observations
<code>imlopts=<i>options</i></code>	IML PROC statement options
<code>init=<i>SAS-data-set</i></code>	initial input experimental design
<code>interact=<i>interaction-list</i></code>	interaction terms
<code>iter=<i>n1</i> &lt; <i>n2</i> &lt; <i>n3</i> &gt;&gt;</code>	maximum number of iterations
<code>levels=<i>value</i></code>	assigning final factor levels
<code>maxdesigns=<i>n</i></code>	maximum number of designs to make
<code>maxiter=<i>n1</i> &lt; <i>n2</i> &lt; <i>n3</i> &gt;&gt;</code>	maximum number of iterations
<code>maxstages=<i>n</i></code>	maximum number of algorithm stages
<code>maxtime=<i>n1</i> &lt; <i>n2</i> &lt; <i>n3</i> &gt;&gt;</code>	approximate maximum run time
<code>mintry=<i>n</i></code>	minimum number of rows to process
<code>mutate=<i>n1</i> &lt; <i>n2</i> &lt; <i>n3</i> &gt;&gt;</code>	mutation probability
<code>mutiter=<i>n1</i> &lt; <i>n2</i> &lt; <i>n3</i> &gt;&gt;</code>	first iteration to consider mutating
<code>n=<i>n</i></code>	number of runs in the design
<code>options=accept</code>	accept designs that violate restrictions
<code>options=check</code>	checks the efficiency of the <code>init=</code> design
<code>options=file</code>	renders the design to a file
<code>options=int</code>	add an intercept variable <code>x0</code> to the design
<code>options=justinit</code>	stop processing after making the initial design
<code>options=largedesign</code>	stop after <code>maxtime=</code> minutes have elapsed
<code>options=lineage</code>	displays the lineage of the orthogonal array
<code>options=nodups</code>	eliminates duplicate runs
<code>options=nofinal</code>	skips displaying the final efficiency
<code>options=nohistory</code>	does not display the iteration history
<code>options=nooadups</code>	check for orthogonal array with duplicate runs
<code>options=noqc</code>	do not use the SAS/QC product
<code>options=nosort</code>	does not sort the design
<code>options=nox</code>	suppress the the <code>xn</code> scalars with restrictions
<code>options=quick</code>	<code>optiter=0, maxdesigns=2, unbalanced=0, tabiter=1</code>
<code>options=quickr</code>	<code>optiter=0, maxdesigns=1, unbalanced=0, tabiter=0</code>
<code>options=quickt</code>	<code>optiter=0, maxdesigns=1, unbalanced=0, tabiter=1</code>
<code>options=render</code>	displays the design compactly in the SAS listing
<code>options=refine</code>	with <code>init=</code> , never reinitializes

Option	Description
<code>options=resrep</code>	reports on the progress of the restrictions
<code>options=+-</code>	renders -1 as - and 1 as + in two-level factors
<code>options=3</code>	applies <code>options=+-</code> to 3-level factors
<code>options=512</code>	adds some designs in 512 runs
<code>optiter=n1 &lt; n2 &gt;</code>	OPTEx iterations
<code>order=value</code>	coordinate exchange column order
<code>out=SAS-data-set</code>	output experimental design
<code>outall=SAS-data-set</code>	output data set with all designs
<code>outeff=SAS-data-set</code>	output data set with final efficiency
<code>outr=SAS-data-set</code>	randomized output experimental design
<code>partial=n</code>	partial-profile design
<code>repeat=n1 n2 n3</code>	times to iterate on a row
<code>reslist=list</code>	constant matrix list
<code>resmac=macro-name</code>	constant matrix creation macro
<code>restrictions=macro-name</code>	restrictions macro
<code>ridge=n</code>	ridging factor
<code>seed=n</code>	random number seed
<code>stopearly=n</code>	the macro can stop early
<code>tabiter=n1 &lt; n2 &gt;</code>	design table initialization iterations
<code>tabsize=n</code>	orthogonal array size
<code>target=n</code>	target efficiency criterion
<code>unbalanced=n1 &lt; n2 &gt;</code>	unbalance initial design iterations

```
%mktex(2 3 ** 7, n=18)
```

```
%mktex(n=104, options=int)
```

```
%mktex(2 ** 5 3 ** 4 5 5 5 6 6, n=60, seed=104, maxtime=.3 .3 .3)
```

```
%macro resmac;
```

```
  client    = {1.29 1.69 2.09 .}[x1];
```

```
  extension = {1.39 1.89 2.39 .}[x2];
```

```
  regional  = {1.99 2.49      .}[x5];
```

```
  private   = {1.49 2.29      .}[x6];
```

```
  national  = {1.99 2.39      .}[x8];
```

```
  navail    = (client  ^= .) + (extension ^= .) + (regional ^= .) +
              (private ^= .) + (national  ^= .);
```

```
  if (navail < 2) | (navail > 4) then bad = abs(navail - 3);
```

```
  else
              bad = 0;
```

```
%mend;
```

```
%mktex( 4 4 2 2 3 3 2 3, n=26, interact=x2*x3 x2*x4 x3*x4 x6*x7,
        restrictions=resmac, seed=377 )
```

# MktKey Macro Options

Option	Description
<code>list</code>	(positional) variable list or n rows and n columns

```
%mktkey(3 5)
```

```
%mktkey(5 10 t)
```

```
%mktkey(x1-x20)
```

# MktLab Macro Options

Option	Description
<code>cfill=character-string</code>	character fill value
<code>data=SAS-data-set</code>	input design data set
<code>dolist=do-list</code>	new values using a do-list syntax
<code>int=variable-list</code>	name of an intercept variable
<code>key=SAS-data-set</code>	key data set
<code>labels=macro-name</code>	macro that provides labels and formats
<code>nfill=number</code>	numeric fill value
<code>options=noprint</code>	suppress the display of the variable mappings
<code>out=SAS-data-set</code>	output data set with recoded design
<code>prefix=variable-prefix</code>	prefix for naming variables
<code>statements=SAS-code</code>	add extra statements
<code>values=value-list</code>	the new values for all of the variables
<code>vars=variable-list</code>	list of variable names

```
%mktex(n=12, options=nosort)
```

```
%mktlab(data=design, values=-1 1, int=Had0, prefix=Had);
```

```
%mktex(n=18, seed=17)
```

```
%mktblock(nblocks=2, factors=x1-x4)
```

```
data key;
  input Brand $ Price Size;
  format price dollar5.2;
  datalines;
Acme 1.49 6
Apex 1.79 8
. 1.99 12
;
```

```
%mktlab(data=blocked, key=key)
```

# MktMDiff Macro Options

Option	Description
layout	(positional) choice data set structure
attrs= <i>macro-variable</i>	macro variable with attribute descriptions
b= <i>b</i>	number of sets (alias for nsets=)
classopts= <i>options-list</i>	class variable options
data= <i>SAS-data-set</i>	input data set
design= <i>SAS-data-set</i>	design data set
group= <i>k</i>	number of groups in the design= data set
k= <i>k</i>	set size (alias for setsize=)
nattrs= <i>t</i>	number of attributes (alias for t=)
nsets= <i>b</i>	number of sets (alias for b=)
options=nocode	just create the OUT= data set
options=noanalysis	just create the OUT= and OUTCODED= data sets
options=nosort	do not sort the parameter estimates table
options=nodrop	do not drop the variable or parameter column
out= <i>SAS-data-set</i>	merged data and design data set
outcoded= <i>SAS-data-set</i>	coded data set
outparm= <i>SAS-data-set</i>	parameter estimates data set
rescale= <i>value</i>	table of rescaled parameter estimates
setsize= <i>k</i>	set size (alias for k=)
t= <i>t</i>	number of attributes or messages (alias for nattrs=)
vars= <i>variable-list</i>	data= data set variables

```
%mktbibd(t=9, k=5, b=18, seed=377, out=sasuser.bibd)
```

```
title "Best Worst Example with Cell Phone Attributes";
```

```
data bestworst;
```

```
input Sub $ @4 (b1-b18 w1-w18) (1.);
```

```
datalines;
```

```
1 188661884399349653941955342212935494
2 765358873891388493922673644336595554
3 782126282892848564995993447213935655
4 481363264246399187162125415351281453
5 787168863811878175225995382352235293
6 787658867891878667495965442313345453
7 788171867736888187465395344393344453
8 788771867711888687445353445353335254
9 188778887896878687425323443242344454
10 788778887816888687445353444343344454
```

```
;
```

```
%let attributes=Camera,Flip,Hands Free,Games,Internet
```

```
,Free Replacement,Battery Life,Large Letters,Applications;
```

```
%phchoice( on )
```

```
%mktmdiff(bw, b=18, t=9, k=5, attrs=attributes,
```

```
data=bestworst, design=sasuser.bibd)
```

# MktMerge Macro Options

Option	Description
<code>blocks=1</code>   <i>variable</i>	blocking variable
<code>data=SAS-data-set</code>	input SAS data set
<code>design=SAS-data-set</code>	input SAS choice design data set
<code>nalts=n</code>	number of alternatives
<code>nsets=n</code>	number of choice sets
<code>out=SAS-data-set</code>	output SAS data set
<code>setvars=variable-list</code>	variables with the data
<code>statements=SAS-statements</code>	additional statements

```
%mktex(3 ** 4, n=18, seed=104)

data results;
  input Subj (choose1-choose18) (1.) @@;
  datalines;
1 333542334333314443 2 333212344333333345 3 333212333333313333
4 133242144334414453 5 335242134333513443 6 333242234333314443
7 333432334332323443 8 333242234334414443 9 333432331352313343
;

data key;
  input Brand $ & Price $;
  datalines;
Brand 1 x1
Brand 2 x2
Brand 3 x3
Brand 4 x4
Another .
;

%mktroll(design=randomized, key=key, alt=brand, out=rolled)

%mktmerge(design=rolled, data=results, out=res2,
          nsets=18, nalts=5, setvars=choose1-choose18)
```

# MktOrth Macro Options

Option	Description
<code>filter=<i>n</i></code>	extra filtering of the design catalog
<code>maxn=<i>n</i></code>	maximum number of runs of interest
<code>maxlev=<i>n</i></code>	maximum number of levels
<code>options=lineage</code>	construct the design lineage
<code>options=mktext</code>	the macro is being called from the %MktEx macro
<code>options=mktruns</code>	the macro is being called from the %MktRuns macro
<code>options=parent</code>	lists only parent designs
<code>options=dups</code>	suppress duplicate and inferior design filtering
<code>options=512</code>	adds some designs in 512 runs
<code>outall=<i>SAS-data-set</i></code>	output data set with all designs
<code>outcat=<i>SAS-data-set</i></code>	design catalog data set
<code>outlev=<i>SAS-data-set</i></code>	output data set with the list of levels
<code>range=<i>range-specification</i></code>	number of runs of interest

```
%mktorth(range=n eq 12 or n eq 18);
```

```
proc print; run;
```

```
%mktorth(range=n=72)
```

```
proc print data=mktdeslev(where=(x2 ge 5 and x3 ge 5 and x4 and x6));
  var n design reference;
run;
```

```
%mktorth(maxn=100, maxlev=6)
```

```
%mktorth(range=n=36, options=lineage)
proc print; run;
```

```
%mktorth(range=n=36, options=parent)
proc print; run;
```

# MktPPro Macro Options

Option	Description
<code>ibd=SAS-data-set</code>	incomplete block design
<code>design=SAS-data-set</code>	orthogonal array
<code>out=SAS-data-set</code>	output partial-profile design
<code>print=print-options</code>	output display options
<code>x=SAS-data-set</code>	incomplete block design incidence matrix

```
%mktex(2 4 2 2 2, n=8)
```

```
data design; set design; drop x2; run;
```

```
%mktbibd(b=20, nattrs=16, setsize=4, seed=104, out=b)
```

```
%mktppro(ibd=b, print=f p)
```

```
%choicelf(data=chdes, model=class(x1-x16 / sta), nsets=80, nalts=2,  
options=relative, beta=zero, init=chdes, initvars=x1-x16)
```

# MktRoll Macro Options

Option	Description
<i>alt=variable</i>	variable with name of each alternative
<i>design=SAS-data-set</i>	input SAS data set
<i>keep=variable-list</i>	factors to keep
<i>key=SAS-data-set</i>	key data set
<i>options=nowarn</i>	suppress the variables not mentioned warning
<i>out=SAS-data-set</i>	output SAS data set
<i>set=variable</i>	choice set number variable

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

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

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

```
proc print; by set; id set; run;
```

# MktRuns Macro Options

Option	Description
<code>list</code>	(positional) numbers of levels of all the factors
<code>interact=<i>interaction-list</i></code>	interaction terms
<code>max=<i>n</i> &lt; <i>m</i> &gt; &lt; <i>f</i> &gt;</code>	largest design sizes to try
<code>n=<i>n</i></code>	design size to evaluate
<code>maxlev=<i>n</i></code>	maximum number of levels
<code>maxoa=<i>n</i></code>	maximum number of orthogonal arrays
<code>options=justparse</code>	used by other Mkt 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=<i>SAS-data-set</i></code>	data set with the suggested sizes
<code>outorth=<i>SAS-data-set</i></code>	data set with orthogonal array list
<code>toobig=<i>n</i></code>	specifies problem that is too big

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

```
%mktruns(2**10 3**3, interact=11|12|13@2)
```

# Paint Macro Options

Option	Description
<code>values=do-list</code>	input data values
<code>var=variable-name</code>	input variable
<code>data=SAS-data-set</code>	input SAS data set
<code>out=SAS-data-set</code>	output SAS data set
<code>macro=macro-name</code>	SAS macro name
<code>colors=&lt;color-list&gt; &lt;data-value-list&gt;</code>	colors list
<code>level=measurement-level</code>	measurement level of the data
<code>symbols=symbols-list</code>	list of symbols
<code>format=SAS-format</code>	nominal and ordinal variable format
<code>order=summary-order</code>	nominal and ordinal variable order
<code>select=n</code>	selection state
<code>show=n</code>	show/hide state
<code>include=n</code>	include/exclude state
<code>label=n</code>	label/unlabel state
<code>rgbround=rounding-list</code>	RGB value rounding instructions
<code>missing=missing-specification</code>	missing value handling
<code>debug=debug-string</code>	debugging information to display

```

data cars(drop=i);
  do i = 1 to 7;
    input (MPG Reliability Acceleration Braking Handling Ride
           Visibility Comfort Quiet Cargo) (1.) +1 @;
    output;
  end;
  datalines;
3334444544 2434453555 2354353545 3244443424 2354553543 3232423224 4145555422
2254541241 2445353555 5335425223 4155555525 2153542242 3333444544 3253353544
;

%paint(values=-1 to 1 by 0.05, macro=setstyle,
       colors=CXEEEEEE red magenta blue cyan white
             cyan blue magenta red CXEEEEEE
             -1 -0.99 -0.75 -0.5 -0.25 0 0.25 0.5 0.75 0.99 1)

proc template;
  edit Base.Corr.StackedMatrix;
    column (RowName RowLabel) (Matrix) * (Matrix2);
  edit matrix;    %setstyle(backgroundcolor)    end;
  end;
run;

ods html body="corr.html" style=statistical;
proc corr data=cars noprob;    run;
ods html close;

proc template;    delete Base.Corr.StackedMatrix;    run;

```

# PHChoice Macro Options

Option	Description
<code>onoff</code>	(positional) <code>on</code> , <code>off</code> , or <code>expb</code>
<code>column</code>	(positional) list of columns

`%phchoice(on)`

`%phchoice(off)`

`%phchoice(expb)`

# PlotIt Macro Options

Option	Description
<code>adjust1=SAS-statements</code>	adjust the preprocessing data set
<code>adjust2=SAS-statements</code>	includes statements with PROC PLOT
<code>adjust3=SAS-statements</code>	extra statements for the final DATA step
<code>adjust4=SAS-statements</code>	extra statements for the final DATA step
<code>adjust5=SAS-statements</code>	extra statements for the final DATA step
<code>antiidea=n</code>	eliminates PREFMAP anti-ideal points
<code>blue=expression</code>	blue part of RGB colors
<code>bright=n</code>	generates random label colors
<code>britypes=type</code>	types to which <code>bright=</code> applies
<code>cframe=color</code>	color of background within the frame
<code>cirsegs=n</code>	circle smoothness parameter
<code>color=color</code>	default color
<code>colors=colors-list</code>	default label and symbol color list
<code>cursegs=n</code>	number of segments in a curve
<code>curvecol=color</code>	color of curve
<code>data=SAS-data-set</code>	input data set
<code>datatype=data-type</code>	data analysis that generated data set
<code>debug=values</code>	debugging output
<code>excolors=color-list</code>	excludes from the Annotate data set
<code>extend=axis-extensions</code>	extend the <i>x</i> and <i>y</i> axes
<code>extraobs=SAS-data-set</code>	extra observations data set
<code>exttypes=type</code>	types for <code>extraobs=</code> data set
<code>filepref=prefix</code>	file name prefix
<code>font=font</code>	default font
<code>framecol=color</code>	color of frame
<code>gdesc=description</code>	catalog description
<code>gname=name</code>	catalog entry
<code>gopplot=goptions</code>	<code>goptions</code> for plotting to screen
<code>gopprint=goptions</code>	<code>goptions</code> for printing
<code>gopts2=goptions</code>	<code>goptions</code> that are always used
<code>gopts=goptions</code>	additional <code>goptions</code>
<code>gout=catalog</code>	<code>proc ganno gout=</code> catalog
<code>green=expression</code>	green part of RGB colors
<code>hminor=n   do-list</code>	horizontal axis minor tick marks
<code>hnoobs=n</code>	horizontal observations for contour plots
<code>hpos=n</code>	horizontal positions in graphics area
<code>href=do-list</code>	horizontal reference lines
<code>hsize=n</code>	horizontal graphics area size
<code>inc=n</code>	<code>haxis=by inc</code> , <code>vaxis=by inc</code>
<code>interpol=method</code>	axis interpolation method
<code>labcol=label-colors</code>	colors for the point labels
<code>label=label-statement</code>	<code>label</code> statement
<code>labelcol=color</code>	color of variable labels
<code>labelvar=label-variable</code>	point label variable
<code>labfont=label-fonts</code>	fonts for the point labels

Option	Description
<code>labsize=label-sizes</code>	sizes for the point labels
<code>ls=n</code>	how line sizes are generated
<code>lsinc=n</code>	increment to line size
<code>lsizes=number-list</code>	line sizes (thicknesses)
<code>makefit=n</code>	proportion of graphics window to use
<code>maxiter=n</code>	maximum number of iterations
<code>maxokpen=n</code>	maximum acceptable penalty sum
<code>method=value</code>	where to send the plot
<code>monochro=color</code>	overrides all other colors
<code>nknots=n</code>	number of knots option
<code>offset=n</code>	move symbols for coincident points
<code>options=border</code>	draws a border box
<code>options=close</code>	close up the border box and the axes
<code>options=diag</code>	draws a diagonal reference line
<code>options=expand</code>	expands the plot to fill the window
<code>options=noback</code>	do not set the frame color
<code>options=nocenter</code>	do not center
<code>options=noclip</code>	do not clip
<code>options=nocode</code>	suppress the PROC PLOT and <code>goptions</code> statements
<code>options=nodelete</code>	do not delete intermediate data sets
<code>options=nohistory</code>	suppress the iteration history table
<code>options=nolegend</code>	suppress the display of the legends
<code>options=noprint</code>	<code>nolegend</code> , <code>nocode</code> , and <code>nohistory</code>
<code>options=square</code>	the same ticks for both axes and a square plot
<code>options=textline</code>	lines overwrite text
<code>out=SAS-data-set</code>	output Annotate data set
<code>outward=none   char</code>	PLOT statement <code>outward=</code>
<code>paint=interpolation</code>	color interpolation
<code>place=placement</code>	generates a <code>placement=</code> option
<code>plotopts=options</code>	PLOT statement options
<code>plotvars=variable-list</code>	<i>y</i> -axis and <i>x</i> -axis variables
<code>post=filename</code>	graphics stream file name
<code>preproc=SAS-data-set</code>	preprocessed <code>data=</code> data set
<code>procopts=options</code>	PROC PLOT statement options
<code>ps=n</code>	page size
<code>radii=do-list</code>	radii of circles
<code>red=expression</code>	red part of RGB colors
<code>regdat=SAS-data-set</code>	intermediate regression results data set
<code>regopts=options</code>	regression curve fitting options
<code>regprint=regression-option</code>	regression options
<code>rgbround=RGB-rounding</code>	<code>paint=</code> rounding factors
<code>rgbtypes=type</code>	types to which <code>paint=</code> and RGB options apply
<code>style=A   B</code>	obsolete option not supported
<code>symbols=symbol-list</code>	plotting symbols
<code>symcol=symbol-colors</code>	colors of the symbols
<code>symfont=symbol fonts</code>	symbol fonts
<code>symlen=n</code>	length of the symbols
<code>symsize=symbol-sizes</code>	sizes of symbols

Option	Description
<code>symtype=</code> <i>symbol-types</i>	types of symbols
<code>symvar=</code> <i>symbol-variable</i>	plotting symbol variable
<code>tempdat1=</code> <i>SAS-data-set</i>	intermediate results data set
<code>tempdat2=</code> <i>SAS-data-set</i>	intermediate results data set
<code>tempdat3=</code> <i>SAS-data-set</i>	intermediate results data set
<code>tempdat4=</code> <i>SAS-data-set</i>	intermediate results data set
<code>tempdat5=</code> <i>SAS-data-set</i>	intermediate results data set
<code>tempdat6=</code> <i>SAS-data-set</i>	intermediate results data set
<code>tickaxes=</code> <i>axis-string</i>	axes to draw tick marks
<code>tickcol=</code> <i>color</i>	color of ticks
<code>tickfor=</code> <i>format</i>	tick format used by <code>interpol=tick</code>
<code>ticklen=</code> <i>n</i>	length of tick mark in horizontal cells
<code>titlecol=</code> <i>color</i>	color of title
<code>tsize=</code> <i>n</i>	default text size
<code>types=</code> <i>observation-types</i>	observations types
<code>typevar=</code> <i>variable</i>	observation types variable
<code>unit=in   cm</code>	<code>hsize=</code> and <code>vsize=</code> unit
<code>vehead=</code> <i>vector-head-size</i>	how to draw vector heads
<code>vminor=</code> <i>n   do-list</i>	vertical axis minor tick marks
<code>vnoobs=</code> <i>n</i>	vertical observations for contour plots
<code>vpos=</code> <i>n</i>	vertical positions in graphics area
<code>vref=</code> <i>do-list</i>	vertical reference lines
<code>vsize=</code> <i>n</i>	vertical graphics area size
<code>vtoh=</code> <i>n</i>	PROC PLOT <code>vtoh=</code> option
<code>xmax=</code> <i>n</i>	maximum horizontal graphics area size
<code>ymax=</code> <i>n</i>	maximum vertical graphics area size

```

data smoke;
  length Employee $ 20;
  input None Light Medium Heavy @13 Employee $ & Weight No Yes;
  label none      = "Nonsmoker"      light   = "Light Smoker"
        medium   = "Medium Smoker"  heavy   = "Heavy Smoker"
        no       = "Nondrinker"     yes     = "Drinker";
  datalines;
  4  2  3  2 Senior Manager   1  0 11
  4  3  7  4 Junior Manager  1  1 17
 25 10 12  4 Senior Employee  1  5 46
 18 24 33 13 Junior Employee  1 10 78
 10  6  7  2 Admin           1  7 18
 42 29 20  9 Nation          -1  .  .
;
proc corresp outc=c;
  var  none -- heavy no yes;
  id   employee;
  weight weight;
  supvar no yes;
run;

%plotit(data=c, datatype=corresp)

```