Chapter Contents Previous Next

The FACTOR Procedure

PROC FACTOR Statement

PROC FACTOR < *options* > **;**

The following new or updated options are available:

ALPHA=*p*
specifies the level of confidence 1-*p* for interval construction. By default, *p* = 0.05, corresponding to 1-*p* = 95% confidence intervals. If *p*
is greater than one, it is interpreted as a percentage and divided by 100. Because the coverage probability is not controlled simultaneously, you may consider supplying a nonconventional *p* using methods such as Bonferroni adjustment.

COVER <**=*p***>
CI <**=*p***>
computes the confidence intervals and optionally specifies the value of factor loading for coverage detection. By default, *p*
= 0. The specified value is represented by an asterisk `*' in the coverage display. This is useful for determining the salience of loadings. For example, if COVER=.4, a display `0*[ ]' indicates that the entire confidence interval is above 0.4, implying strong evidence for the salience of the loading. See the section "Confidence Intervals and the Salience of Factor Loadings" for more details.

HKPOWER=*p*
HKP=*p*
specifies the power of the square roots of the eigenvalues used to rescale the eigenvectors for Harris-Kaiser (ROTATE=HK) rotation, assuming that the factors are extracted by the principal factor method. If the principal factor method is not used for factor extraction, the eigenvectors are replaced by the normalized columns of the unrotated factor matrix, and the eigenvalues replaced by the column normalizing constants. HKPOWER= values between 0.0 and 1.0 are reasonable. The default value is 0.0, yielding the independent cluster solution, in which each variable tends to have a large loading on only one factor. An HKPOWER= value of 1.0 is equivalent to an orthogonal rotation, with the varimax rotation as the default. You can also specify the HKPOWER= option with ROTATE=QUARTIMAX, ROTATE=BIQUARTIMAX, ROTATE=EQUAMAX, or ROTATE=ORTHOMAX, and so on. The only restriction is that the Harris-Kaiser rotation must be associated with an orthogonal rotation.

PREROTATE=*name*
PRE=*name*
specifies the prerotation method for the option ROTATE=PROMAX. Any rotation method other than PROMAX or PROCRUSTES can be used. See the ROTATE= option for the available prerotation methods. The default is PREROTATE=VARIMAX. If a previously rotated pattern is read using the option METHOD=PATTERN, you should specify the PREROTATE=NONE option.

RCONVERGE=*p*
RCONV=*p*
specifies the convergence criterion for rotation cycles. Rotation stops when the scaled change of the simplicity function value is less than the RCONVERGE= value. The default convergence criterion is

$$|f_{new} - f_{old}|/K < \epsilon$$

where f_{new} and f_{old} are simplicity function values of the current cycle and the previous cycle, respectively, $K=max(1,|f_{old}|)$ is a scaling factor, and ϵ is 1E-9 by default and is modified by the RCONVERGE= value.

RITER=*n*
specifies the maximum number of cycles for factor rotation. Except for promax and Procrustes, you can use the RITER= option with all rotation methods. The default is the maximum between 100 and ten times of the number of variables.

ROTATE=*name*
R=*name*
specifies the rotation method. The default is ROTATE=NONE.

Valid *name*s for orthogonal rotations are as follows:

BIQUARTIMAX | BIQMAX
specifies orthogonal biquartimax rotation. This corresponds to the specification ROTATE=ORTHOMAX(.5).

EQUAMAX | E
specifies orthogonal equamax rotation. This corresponds to the specification ROTATE=ORTHOMAX with GAMMA=*number of factors*/2.

FACTORPARSIMAX | FPA
specifies orthogonal factor parsimax rotation. This corresponds to the specification ROTATE=ORTHOMAX with GAMMA=*number of variables*.

NONE | N
specifies that no rotation be performed, leaving the original orthogonal solution.

ORTHCF(*p1*,*p2*) | ORCF(*p1*,*p2*)
specifies the orthogonal Crawford-Ferguson rotation with the weights *p1* and *p2* for variable and factor parsimony, respectively. See the definitions of weights in the section "Simplicity Functions for Rotations".

ORTHGENCF(*p1*,*p2*,*p3*,*p4*) | ORGENCF(*p1*,*p2*,*p3*,*p4*)
specifies the orthogonal generalized Crawford-Ferguson rotation with the four weights *p1*, *p2*, *p3*, and *p4*. See the definitions of weights in the section "Simplicity Functions for Rotations".

ORTHOMAX<(*p*)> | ORMAX<(*p*)>
specifies the orthomax rotation. If ROTATE=ORTHOMAX is used, the orthomax weight is specified by the GAMMA= option. You can also specify the GAMMA= value in the parentheses of ROTATE=ORTHOMAX(*p*). See the definition of the orthomax weight in the section "Simplicity Functions for Rotations".

PARSIMAX | PA
specifies orthogonal parsimax rotation. This corresponds to the specification ROTATE=ORTHOMAX with

GAMMA = [(nvar ×(nfact - 1))/(nvar + nfact - 2)]

where *nvar* is the number of variables, and *nfact* is the number of factors.

QUARTIMAX | QMAX | Q
specifies orthogonal quartimax rotation. This corresponds to the specification ROTATE=ORTHOMAX(0).

VARIMAX | V
specifies orthogonal varimax rotation. This corresponds to the specification ROTATE=ORTHOMAX with GAMMA=1.

Valid *name*s for oblique rotations are as follows:

BIQUARTIMIN | BIQMIN
specifies biquartimin rotation. It corresponds to the specification ROTATE=OBLIMIN(.5) or ROTATE=OBLIMIN with TAU=.5.

COVARIMIN | CVMIN
specifies covarimin rotation. It corresponds to the specification ROTATE=OBLIMIN(1) or ROTATE=OBLIMIN with TAU=1.

HK<(*p*)> | H<(*p*)>
specifies Harris-Kaiser case II orthoblique rotation. When specifying this option, you can use the HKPOWER= option to set the power of the square roots of the eigenvalues by which the eigenvectors are scaled, assuming that the factors are extracted by the principal factor method. For other extraction methods, the unrotated factor pattern is column normalized. The power is then applied to the column normalizing constants, instead of the eigenvalues. You can also use ROTATE=HK(*p*), with *p* representing the HKPOWER= value. The default associated orthogonal rotation with ROTATE=HK is the varimax rotation without Kaiser normalization. You may associate the Harris-Kaiser with other orthogonal rotations using the ROTATE= option together with the HKPOWER= option.

OBBIQUARTIMAX | OBIQMAX
specifies oblique biquartimax rotation.

OBEQUAMAX | OE
specifies oblique equamax rotation.

OBFACTORPARSIMAX | OFPA
specifies oblique factor parsimax rotation.

OBLICF(*p1*,*p2*) | OBCF(*p1*,*p2*)
specifies the oblique Crawford-Ferguson rotation with the weights *p1* and *p2* for variable and factor parsimony, respectively. See the definitions of weights in the section "Simplicity Functions for Rotations".

OBLIGENCF(*p1*,*p2*,*p3*,*p4*) | OBGENCF(*p1*,*p2*,*p3*,*p4*)
specifies the oblique generalized Crawford-Ferguson rotation with the four weights *p1*, *p2*, *p3*, and *p4*. See the definitions of weights in the section "Simplicity Functions for Rotations".

OBLIMIN<(*p*)> | OBMIN<(*p*)>
specifies the oblimin rotation. If ROTATE=OBLIMIN is used, the oblimin weight is specified by the TAU= option. Alternatively, ROTATE=OBLIMIN(*p*) specifies *p* as the TAU= value. See the definition of the oblimin weight in the section "Simplicity Functions for Rotations".

OBPARSIMAX | OPA
specifies oblique parsimax rotation.

OBQUARTIMAX | OQMAX
specifies oblique quartimax rotation. This is the same as the QUARTIMIN method.

OBVARIMAX | OV
specifies oblique varimax rotation.

PROCRUSTES
specifies oblique Procrustes rotation with the target pattern provided by the TARGET= data set. The unrestricted least squares method is used with factors scaled to unit variance after rotation.

PROMAX<(*p*)> | P<(*p*)>
specifies oblique promax rotation. You can use the PREROTATE= option to set the desirable prerotation method, orthogonal or oblique. When using with ROTATE=PROMAX, the POWER= option lets you specify the power for forming the target. You can also use ROTATE=PROMAX(*p*), where *p* represents the POWER= value.

QUARTIMIN | QMIN
specifies quartimin rotation. It is the same as the oblique quartimax method. It also corresponds to the specification ROTATE=OBLIMIN(0) or ROTATE=OBLIMIN with TAU=0.

SE
STDERR
computes standard errors for various classes of unrotated and rotated solutions under the maximum likelihood estimation.


Chapter Contents Previous Next Top