

Postestimation commands
Stored results

[predict](#)
Methods and formulas

[estat](#)
References

Remarks and examples
[Also see](#)

Postestimation commands

The following postestimation commands are of special interest after `factor` and `factormat`:

| Command | Description |
|----------------------------------|---|
| <code>estat anti</code> | anti-image correlation and covariance matrices |
| <code>estat common</code> | correlation matrix of the common factors |
| <code>estat factors</code> | AIC and BIC model-selection criteria for different numbers of factors |
| <code>estat kmo</code> | Kaiser–Meyer–Olkin measure of sampling adequacy |
| <code>estat residuals</code> | matrix of correlation residuals |
| <code>estat rotatecompare</code> | compare rotated and unrotated loadings |
| <code>estat smc</code> | squared multiple correlations between each variable and the rest |
| <code>estat structure</code> | correlations between variables and common factors |
| * <code>estat summarize</code> | estimation sample summary |
| <code>loadingplot</code> | plot factor loadings |
| <code>rotate</code> | rotate factor loadings |
| <code>scoreplot</code> | plot score variables |
| <code>screepplot</code> | plot eigenvalues |

*`estat summarize` is not available after `factormat`.

The following standard postestimation commands are also available:

| Command | Description |
|--------------------------|--------------------------------|
| * <code>estimates</code> | cataloging estimation results |
| † <code>predict</code> | regression and Bartlett scores |

*`estimates table` is not allowed, and `estimates stats` is allowed only with the `ml` factor method.

†`predict` after `factormat` works only if you have variables in memory that match the names specified in `factormat`.
`predict` assumes mean zero and standard deviation one unless the `means()` and `sds()` options of `factormat` were provided.

predict

Description for predict

predict creates new variables containing predictions such as factors scored by the regression method or by the Bartlett method.

Menu for predict

Statistics > Postestimation

Syntax for predict

```
predict [type] {stub*|newvarlist} [if] [in] [, statistic options]
```

| statistic | Description |
|------------|--|
| Main | |
| regression | regression scoring method; the default |
| bartlett | Bartlett scoring method |

| options | Description |
|--------------|--|
| Main | |
| norotated | use unrotated results, even when rotated results are available |
| notable | suppress table of scoring coefficients |
| format(%fmt) | format for displaying the scoring coefficients |

Options for predict

Main

regression produces factors scored by the regression method. This is the default.

bartlett produces factors scored by the method suggested by Bartlett (1937, 1938). This method produces unbiased factors, but they may be less accurate than those produced by the default regression method suggested by Thomson (1951). Regression-scored factors have the smallest mean squared error from the true factors but may be biased.

norotated specifies that unrotated factors be scored even when you have previously issued a rotate command. The default is to use rotated factors if they are available and unrotated factors otherwise.

notable suppresses the table of scoring coefficients.

format(%fmt) specifies the display format for scoring coefficients.

estat

Description for estat

`estat anti` displays the anti-image correlation and anti-image covariance matrices. These are minus the partial covariance and minus the partial correlation matrices of all pairs of variables, holding all other variables constant.

`estat common` displays the correlation matrix of the common factors. For orthogonal factor loadings, the common factors are uncorrelated, and hence an identity matrix is shown. `estat common` is of more interest after oblique rotations.

`estat factors` displays model-selection criteria (AIC and BIC) for models with 1, 2, ..., # factors. Each model is estimated using maximum likelihood (that is, using the `ml` option of `factor`).

`estat kmo` specifies that the Kaiser–Meyer–Olkin (KMO) measure of sampling adequacy be displayed. KMO takes values between 0 and 1, with small values meaning that overall the variables have too little in common to warrant a factor analysis. Historically, the following labels are given to values of KMO ([Kaiser 1974](#)):

| | |
|--------------|--------------|
| 0.00 to 0.49 | unacceptable |
| 0.50 to 0.59 | miserable |
| 0.60 to 0.69 | mediocre |
| 0.70 to 0.79 | middling |
| 0.80 to 0.89 | meritorious |
| 0.90 to 1.00 | marvelous |

`estat residuals` displays the raw or standardized residuals of the observed correlations with respect to the fitted (reproduced) correlation matrix.

`estat rotatecompare` displays the unrotated factor loadings and the most recent rotated factor loadings.

`estat smc` displays the squared multiple correlations between each variable and all other variables. SMC is a theoretical lower bound for communality, so it is an upper bound for uniqueness. The `pf` factor method estimates the communalities by `smc`.

`estat structure` displays the factor structure, that is, the correlations between the variables and the common factors.

`estat summarize` displays summary statistics of the variables in the factor analysis over the estimation sample. This subcommand is, of course, not available after `factormat`.

Menu for estat

Statistics > Postestimation

Syntax for estat

Anti-image correlation/covariance matrices

```
estat anti [ , nocorr nocov format(%fmt) ]
```

Correlation of common factors

```
estat common [ , norotated format(%fmt) ]
```

Model-selection criteria

```
estat factors [ , factors(#) detail ]
```

Sample adequacy measures

```
estat kmo [ , novar format(%fmt) ]
```

Residuals of correlation matrix

```
estat residuals [ , fitted obs sresiduals format(%fmt) ]
```

Comparison of rotated and unrotated loadings

```
estat rotatecompare [ , format(%fmt) ]
```

Squared multiple correlations

```
estat smc [ , format(%fmt) ]
```

Correlations between variables and common factors

```
estat structure [ , norotated format(%fmt) ]
```

Summarize variables for estimation sample

```
estat summarize [ , labels noheader noweights ]
```

collect is allowed with estat anti, estat common, estat factors, estat kmo, estat residuals, estat smc, estat structure, and estat summarize; see [\[U\] 11.1.10 Prefix commands](#).

Options for estat

Main

nocorr, an option used with estat anti, suppresses the display of the anti-image correlation matrix.

nocov, an option used with estat anti, suppresses the display of the anti-image covariance matrix.

format(%fmt) specifies the display format. The defaults differ between the subcommands.

norotated, an option used with estat common and estat structure, requests that the displayed and returned results be based on the unrotated original factor solution rather than on the last rotation (orthogonal or oblique).

`factors(#)`, an option used with `estat factors`, specifies the maximum number of factors to include in the summary table.

`detail`, an option used with `estat factors`, presents the output from each run of `factor` (or `factormat`) used in the computations of the AIC and BIC values.

`novar`, an option used with `estat kmo`, suppresses the KMO measures of sampling adequacy for the variables in the factor analysis, displaying the overall KMO measure only.

`fitted`, an option used with `estat residuals`, displays the fitted (reconstructed) correlation matrix on the basis of the retained factors.

`obs`, an option used with `estat residuals`, displays the observed correlation matrix.

`sresiduals`, an option used with `estat residuals`, displays the matrix of standardized residuals of the correlations. Be careful when interpreting these residuals; see [Jöreskog and Sörbom \(1988\)](#).

`labels`, `noheader`, and `noweights` are the same as for the generic `estat summarize` command; see [\[R\] estat summarize](#).

Remarks and examples

Remarks are presented under the following headings:

Postestimation statistics
Plots of eigenvalues, factor loadings, and scores
Rotating the factor loadings
Factor scores

Postestimation statistics

Many postestimation statistics are available after `factor` and `factormat`.

► Example 1: Squared multiple correlations

After `factor` and `factormat` there are several “classical” methods for assessing whether the variables have enough in common to have warranted the use of a factor model. One method is to examine the squared multiple correlations of each variable with all other variables—this is usually an upper bound to *communality* and thus a lower bound to $1 - \text{uniqueness}(= \text{communality})$ of the variables.

```
. use https://www.stata-press.com/data/r19/bg2
(Physician-cost data)

. quietly factor bg2cost1-bg2cost6, factors(2) ml

. estat smc

Squared multiple correlations of variables with all other variables
```

| Variable | smc |
|----------|--------|
| bg2cost1 | 0.1054 |
| bg2cost2 | 0.1370 |
| bg2cost3 | 0.1637 |
| bg2cost4 | 0.0866 |
| bg2cost5 | 0.1671 |
| bg2cost6 | 0.1683 |

Other diagnostic tools, such as examining the anti-image correlation and anti-image covariance matrices (`estat anti`) and the Kaiser–Meyer–Olkin measure of sampling adequacy (`estat kmo`), are also available. See [MV] [pca postestimation](#) for an illustration of their use.



► Example 2: Model-selection criteria

Another set of postestimation tools help in determining the number of factors that should be retained. Later we will show the use of `screeplot` for producing a scree plot—a plot of the explained variance by the common factors. This is often used as a visual guide for selecting the number of factors to retain.

Some authors advocate the standard model information criteria AIC and BIC for determining the number of factors (Schwarz 1978; Akaike 1987). This presupposes that the factors are extracted by maximum likelihood. `estat factors` provides these measures.

```
. estat factors
Factor analysis with different numbers of factors (maximum likelihood)
```

| #factors | loglik | df_m | df_r | AIC | BIC |
|----------|-----------|------|------|----------|----------|
| 1 | -60.53727 | 6 | 9 | 133.0745 | 159.1273 |
| 2 | -6.842448 | 11 | 4 | 35.6849 | 83.44823 |
| 3 | -3.34e-12 | 15 | 0 | 30 | 95.13182 |

no Heywood cases encountered

The table shows the AIC and BIC statistics for the models with 1, 2, and 3 factors. The three-factor model is saturated, with 0 degrees of freedom. In this trivial case, and excluding the saturated case, both criteria select the two-factor model.



► Example 3: Structure matrix and observed correlations

Two `estat` subcommands display statistics that help in interpreting the model and the results—in particular after an oblique rotation. `estat structure` displays the *structure* matrix containing the correlations between the (manifest) variables and the common factors.

```
. estat structure
Structure matrix: correlations between variables and common factors
```

| Variable | Factor1 | Factor2 |
|----------|---------|---------|
| bg2cost1 | -0.1371 | 0.4235 |
| bg2cost2 | 0.4140 | 0.1994 |
| bg2cost3 | 0.6199 | 0.3692 |
| bg2cost4 | 0.3577 | 0.0909 |
| bg2cost5 | -0.3752 | 0.4355 |
| bg2cost6 | -0.4295 | 0.4395 |

This matrix of correlations coincides with the pattern matrix, that is, the matrix with factor loadings. This holds true for the unrotated factor solution as well as after an orthogonal rotation, such as a varimax rotation. It does not hold true after an oblique rotation. After an oblique rotation, the common factors are correlated. This correlation between the common factors also influences the correlation between the

common factors and the manifest variables. The correlation matrix of the common factors is displayed by the common subcommand of `estat`. Because we have not yet rotated, we would see only an identity matrix. Later we show `estat common` output after an oblique rotation.

To assess the quality of a factor model, we may compare the observed correlation matrix \mathbf{C} with the fitted (“reconstructed”) matrix $\widehat{\Sigma} = \widehat{\Lambda}\widehat{\Phi}\widehat{\Lambda}' + \widehat{\Psi}$ by examining the raw residuals $\mathbf{C} - \widehat{\Sigma}$.

```
. estat residuals, obs fit
```

Observed correlations

| Variable | bg2co~1 | bg2co~2 | bg2co~3 | bg2co~4 | bg2co~5 | bg2co~6 |
|----------|---------|---------|---------|---------|---------|---------|
| bg2cost1 | 1.0000 | | | | | |
| bg2cost2 | 0.0920 | 1.0000 | | | | |
| bg2cost3 | 0.0540 | 0.3282 | 1.0000 | | | |
| bg2cost4 | -0.0380 | 0.1420 | 0.2676 | 1.0000 | | |
| bg2cost5 | 0.2380 | -0.1394 | -0.0550 | -0.0567 | 1.0000 | |
| bg2cost6 | 0.2431 | -0.0671 | -0.1075 | -0.1329 | 0.3524 | 1.0000 |

Fitted ("reconstructed") values for correlations

| Variable | bg2co~1 | bg2co~2 | bg2co~3 | bg2co~4 | bg2co~5 | bg2co~6 |
|----------|---------|---------|---------|---------|---------|---------|
| bg2cost1 | 1.0000 | | | | | |
| bg2cost2 | 0.0277 | 1.0000 | | | | |
| bg2cost3 | 0.0714 | 0.3303 | 0.9999 | | | |
| bg2cost4 | -0.0106 | 0.1662 | 0.2553 | 1.0000 | | |
| bg2cost5 | 0.2359 | -0.0685 | -0.0718 | -0.0946 | 1.0000 | |
| bg2cost6 | 0.2450 | -0.0902 | -0.1040 | -0.1137 | 0.3525 | 1.0000 |

Raw residuals of correlations (observed-fitted)

| Variable | bg2co~1 | bg2co~2 | bg2co~3 | bg2co~4 | bg2co~5 | bg2co~6 |
|----------|---------|---------|---------|---------|---------|---------|
| bg2cost1 | -0.0000 | | | | | |
| bg2cost2 | 0.0643 | -0.0000 | | | | |
| bg2cost3 | -0.0174 | -0.0021 | 0.0001 | | | |
| bg2cost4 | -0.0274 | -0.0242 | 0.0124 | -0.0000 | | |
| bg2cost5 | 0.0021 | -0.0709 | 0.0168 | 0.0379 | 0.0000 | |
| bg2cost6 | -0.0019 | 0.0231 | -0.0035 | -0.0193 | -0.0002 | -0.0000 |

To gauge the size of the residuals, `estat residuals` can also display the standardized residuals.

```
. estat residuals, sres
```

Standardized residuals of correlations

| Variable | bg2co~1 | bg2co~2 | bg2co~3 | bg2co~4 | bg2co~5 | bg2co~6 |
|----------|---------|---------|---------|---------|---------|---------|
| bg2cost1 | -0.0001 | | | | | |
| bg2cost2 | 1.5324 | -0.0003 | | | | |
| bg2cost3 | -0.4140 | -0.0480 | 0.0011 | | | |
| bg2cost4 | -0.6538 | -0.5693 | 0.2859 | -0.0000 | | |
| bg2cost5 | 0.0484 | -1.6848 | 0.3993 | 0.9003 | 0.0001 | |
| bg2cost6 | -0.0434 | 0.5480 | -0.0836 | -0.4560 | -0.0037 | -0.0000 |

Be careful when interpreting these standardized residuals, as they tend to be smaller than normalized residuals; that is, these residuals tend to have a smaller variance than 1 if the model is true (see [Bollen \[1989\]](#)).



Plots of eigenvalues, factor loadings, and scores

Scree plots, factor loading plots, and score plots are easily obtained after `factor` and `factormat`.

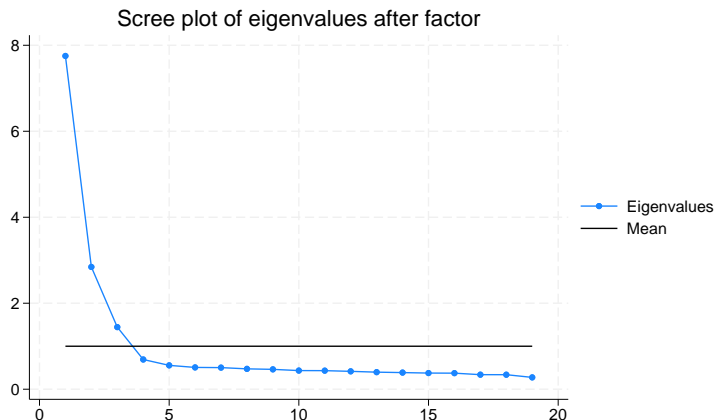
► Example 4: The scree plot

The scree plot is a popular tool for determining the number of factors to be retained. A scree plot is a plot of the eigenvalues shown in decreasing order ([Cattell 1966](#)). We fit a factor model, extracting factors with the principal factor method.

```
. use https://www.stata-press.com/data/r19/sp2
. factor ghp31-ghp05, pcf
  (output omitted)
```

How many factors should we retain? We issue the `screeplot` command with the `mean` option, specifying that a horizontal line be plotted at the mean of the eigenvalues (a height of 1 because we are dealing with the eigenvalues of a correlation matrix).

```
. screeplot, mean
```



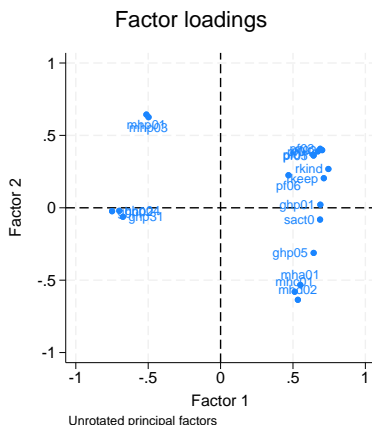
The plot suggests that we retain three factors, both because of the shape of the scree plot and because of Kaiser's well-known criterion suggesting that we retain factors with eigenvalue larger than 1. We may specify the option `mineigen(1)` during estimation to enforce this criterion. Here there is no need—`mineigen(1)` is the default with `pcf`.



► Example 5: Factor loadings plot

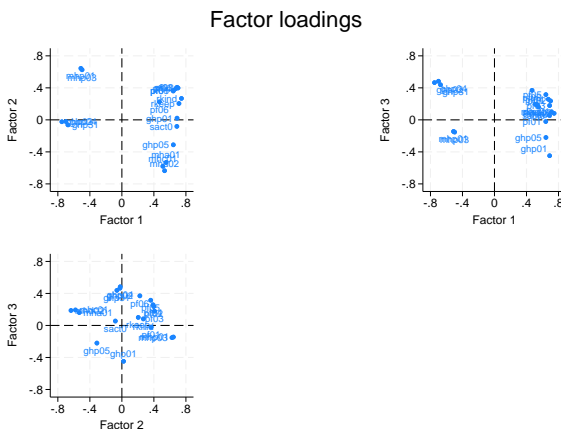
A second plot that is sometimes useful is the factor loadings plot. We display the plot with the loadings of the leading two factors.

```
. loadingplot, xline(0) yline(0) aspect(1) note(Unrotated principal factors)
```



The plot makes it relatively easy to identify clusters of variables with similar loadings. With more than two factors, we can choose to see the multiple plots in a matrix style or a combined-graph style. The default is matrix style, but the combined style allows better control over various graph options—for instance, the addition of `xline(0)` and `yline(0)`. Here is a combined style graph.

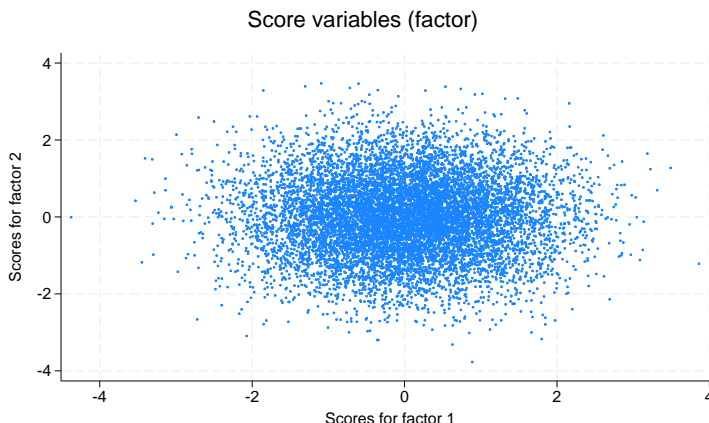
```
. loadingplot, factors(3) combined xline(0) yline(0) aspect(1)
> xlabel(-0.8(0.4)0.8) ylabel(-0.8(0.4)0.8)
```



► Example 6: Score variables plot

Common factor scores can also be plotted for the observations by using the `scoreplot` command. (See the discussion of [predict](#) to see how you can produce score variables.)

```
. scoreplot, msymbol(smcircle) msize(tiny)
```



With so many observations, the plot's main purpose is to identify extreme cases. With smaller datasets with meaningful descriptions of the observations (for example, country names, brands), the score plot is good for visually clustering observations with similar loadings.



See [\[MV\] scoreplot](#) for more examples of `loadingplot` and `scoreplot`.

□ Technical note

The loading plots and score plots we have shown were for the original unrotated factor solution. After rotating (which we will discuss next), these plots display the most recent rotated solution. Specify option `norotated` to refer to the unrotated result. To display the plots of rotated and unrotated results at the same time, you may use either of the following two approaches. First, you may display them in different Graph windows.

```
. plotcmd, norotated name(name1)
. plotcmd, name(name2)
```

Alternatively, you may save the plots and create a combined graph

```
. plotcmd, norotated saving(name1)
. plotcmd, saving(name2)
. graph combine name1.gph name2.gph
```

See [\[G-2\] graph combine](#) for details.



Rotating the factor loadings

Rotation is an attempt to describe the information in several factors by reexpressing them so that loadings on a few variables are as large as possible, and loadings on the rest of the variables are as small as possible. We have this freedom to reexpress because of the indeterminant nature of the factor model. For example, if you find that z_1 and z_2 are two factors, then $z_1 + z_2$ and $z_1 - z_2$ are equally valid solutions.

□ Technical note

Said more technically: we are trying to find a set of f factor variables such that the observed variables can be best explained by regressing them on the f factor variables. Usually, f is a small number such as 1 or 2. If $f \geq 2$, there is an inherent indeterminacy in the construction of the factors because any linear combination of the calculated factors serves equally well as a set of regressors. Rotation capitalizes on this indeterminacy to create a set of variables that looks as much like the original variables as possible.

□

The `rotate` command modifies the results of the last `factor` or `factormat` command to create a set of loadings that are more interpretable than those produced by `factor` or `factormat`. You may perform one factor analysis followed by several `rotate` commands, thus experimenting with different types of rotation. If you retain too few factors, the variables for several distinct concepts may be merged, as in our [example](#) below. If you retain too many factors, several factors may attempt to measure the same concept, causing the factors to get in each other's way, suggesting too many distinct concepts after rotation.

□ Technical note

It is possible to restrict rotation to a number of leading factors. For instance, if you extracted three factors, you may specify the option `factors(2)` to `rotate` to exclude the third factor from being rotated. The new two leading factors are combinations of the initial two leading factors and are not affected by the fixed factor.

□

► Example 7: Orthogonal varimax rotation

We return to our physician-cost [example](#) in [\[MV\] factor](#) and perform a factor analysis using the principal-component factor method, retaining two factors. We then tell `rotate` to apply the default orthogonal varimax rotation ([Kaiser 1958](#)).

```
. use https://www.stata-press.com/data/r19/bg2, clear
(Physician-cost data)

. quietly factor bg2cost1-bg2cost6, pcf factors(2)

. rotate
```

| | | | |
|---|------------------|---|-----|
| Factor analysis/correlation | Number of obs | = | 568 |
| Method: principal-component factors | Retained factors | = | 2 |
| Rotation: orthogonal varimax (Kaiser off) | Number of params | = | 11 |

| Factor | Variance | Difference | Proportion | Cumulative |
|---------|----------|------------|------------|------------|
| Factor1 | 1.57170 | 0.03430 | 0.2619 | 0.2619 |
| Factor2 | 1.53740 | . | 0.2562 | 0.5182 |

LR test: independent vs. saturated: $\chi^2(15) = 269.07$ Prob> $\chi^2 = 0.0000$

Rotated factor loadings (pattern matrix) and unique variances

| Variable | Factor1 | Factor2 | Uniqueness |
|----------|---------|---------|------------|
| bg2cost1 | 0.6853 | 0.2300 | 0.4775 |
| bg2cost2 | -0.0126 | 0.7142 | 0.4898 |
| bg2cost3 | -0.0161 | 0.7818 | 0.3886 |
| bg2cost4 | -0.1502 | 0.5703 | 0.6521 |
| bg2cost5 | 0.7292 | -0.1198 | 0.4539 |
| bg2cost6 | 0.7398 | -0.1537 | 0.4290 |

Factor rotation matrix

| | Factor1 | Factor2 |
|---------|---------|---------|
| Factor1 | 0.7460 | -0.6659 |
| Factor2 | 0.6659 | 0.7460 |

Here the factors are rotated so that the three “negative” items are grouped together and the three “positive” items are grouped.

Look at the uniqueness column. *Uniqueness* is the percentage of variance for the variable that is not explained by the common factors; we may also think of it as the variances of the specific factors for the variables. We stress that rotation involves the “common factors”, so the *uniqueness* is not affected by the rotation. As we noted in [\[MV\] factor](#), the uniqueness is relatively high in this example, placing doubt on the usefulness of the factor model here.

► Example 8: More orthogonal varimax rotation

Here we examine 19 variables describing various aspects of health. These variables were collected from a random selection of 9,999 visitors to doctors' offices by [Tarlov et al. \(1989\)](#). Factor analysis yields three clear factors. We then examine several rotations of these three factors.

```
. use https://www.stata-press.com/data/r19/sp2
. describe
```

```
Contains data from https://www.stata-press.com/data/r19/sp2.dta
Observations:      9,999
Variables:         20                26 Jan 2024 09:26
                                   (_dta has notes)
```

| Variable name | Storage type | Display format | Value label | Variable label |
|---------------|--------------|----------------|-------------|---|
| patid | int | %9.0g | | Case ID |
| ghp31 | float | %9.0g | | Health excellent, very good, good, fair, poor |
| pf01 | float | %9.0g | | How long limit vigorous activity |
| pf02 | float | %9.0g | | How long limit moderate activity |
| pf03 | float | %9.0g | | How long limit walk/climb |
| pf04 | float | %9.0g | | How long limit bend/stoop |
| pf05 | float | %9.0g | | How long limit walk 1 block |
| pf06 | float | %9.0g | | How long limit eat/dress/bath |
| rkeep | float | %9.0g | | Does health keep work-job-hse |
| rkind | float | %9.0g | | Can't do kind/amount of work |
| sact0 | float | %9.0g | | Last month limit activities |
| mha01 | float | %9.0g | | Last month very nervous |
| mhp03 | float | %9.0g | | Last month calm/peaceful |
| mhd02 | float | %9.0g | | Last month downhearted/blue |
| mhp01 | float | %9.0g | | Last month a happy person |
| mhc01 | float | %9.0g | | Last month down in the dumps |
| ghp01 | float | %9.0g | | Somewhat ill |
| ghp04 | float | %9.0g | | Healthy as anybody I know |
| ghp02 | float | %9.0g | | Health is excellent |
| ghp05 | float | %9.0g | | Feel bad lately |

Sorted by: patid

We now perform our factorization, requesting that three factors be retained.

```
. factor ghp31-ghp05, factors(3)
(obs=9,999)
```

| | | | |
|-----------------------------|------------------|---|-------|
| Factor analysis/correlation | Number of obs | = | 9,999 |
| Method: principal factors | Retained factors | = | 3 |
| Rotation: (unrotated) | Number of params | = | 54 |

| Factor | Eigenvalue | Difference | Proportion | Cumulative |
|----------|------------|------------|------------|------------|
| Factor1 | 7.27086 | 4.90563 | 0.7534 | 0.7534 |
| Factor2 | 2.36523 | 1.38826 | 0.2451 | 0.9985 |
| Factor3 | 0.97697 | 1.00351 | 0.1012 | 1.0997 |
| Factor4 | -0.02654 | 0.00538 | -0.0027 | 1.0970 |
| Factor5 | -0.03191 | 0.00378 | -0.0033 | 1.0937 |
| Factor6 | -0.03569 | 0.00353 | -0.0037 | 1.0900 |
| Factor7 | -0.03922 | 0.00271 | -0.0041 | 1.0859 |
| Factor8 | -0.04193 | 0.00662 | -0.0043 | 1.0815 |
| Factor9 | -0.04855 | 0.01015 | -0.0050 | 1.0765 |
| Factor10 | -0.05870 | 0.00250 | -0.0061 | 1.0704 |
| Factor11 | -0.06120 | 0.00224 | -0.0063 | 1.0641 |
| Factor12 | -0.06344 | 0.00376 | -0.0066 | 1.0575 |
| Factor13 | -0.06720 | 0.00345 | -0.0070 | 1.0506 |
| Factor14 | -0.07065 | 0.00185 | -0.0073 | 1.0432 |
| Factor15 | -0.07250 | 0.00033 | -0.0075 | 1.0357 |
| Factor16 | -0.07283 | 0.00772 | -0.0075 | 1.0282 |
| Factor17 | -0.08055 | 0.01190 | -0.0083 | 1.0198 |
| Factor18 | -0.09245 | 0.00649 | -0.0096 | 1.0103 |
| Factor19 | -0.09894 | . | -0.0103 | 1.0000 |

LR test: independent vs. saturated: $\chi^2(171) = 1.0e+05$ Prob> $\chi^2 = 0.0000$

Factor loadings (pattern matrix) and unique variances

| Variable | Factor1 | Factor2 | Factor3 | Uniqueness |
|----------|---------|---------|---------|------------|
| ghp31 | -0.6519 | -0.0562 | 0.3440 | 0.4535 |
| pf01 | 0.6150 | 0.3226 | -0.0072 | 0.5177 |
| pf02 | 0.6867 | 0.3737 | 0.2175 | 0.3415 |
| pf03 | 0.6712 | 0.3774 | 0.1621 | 0.3807 |
| pf04 | 0.6540 | 0.3588 | 0.2268 | 0.3921 |
| pf05 | 0.6209 | 0.3258 | 0.2631 | 0.4392 |
| pf06 | 0.4370 | 0.1803 | 0.2241 | 0.7263 |
| rkeep | 0.6868 | 0.1820 | 0.0870 | 0.4876 |
| rkind | 0.7244 | 0.2464 | 0.0780 | 0.4085 |
| sact0 | 0.6556 | -0.0719 | 0.0461 | 0.5628 |
| mha01 | 0.5297 | -0.4773 | 0.1268 | 0.4755 |
| mhp03 | -0.4810 | 0.5691 | -0.1238 | 0.4294 |
| mhd02 | 0.5208 | -0.5949 | 0.1623 | 0.3485 |
| mhp01 | -0.4980 | 0.5955 | -0.1225 | 0.3824 |
| mhc01 | 0.4927 | -0.5215 | 0.1531 | 0.4618 |
| ghp01 | 0.6686 | 0.0194 | -0.3621 | 0.4215 |
| ghp04 | -0.6833 | -0.0195 | 0.4089 | 0.3656 |
| ghp02 | -0.7398 | -0.0227 | 0.4212 | 0.2748 |
| ghp05 | 0.6163 | -0.2760 | -0.1626 | 0.5175 |

The first factor is a general health factor. (To understand that claim, compare the factor loadings with the description of the variables as shown by `describe` above. Also, just as with the physician-cost data, the sense of some of the coded responses is reversed.) The second factor loads most highly on the five “men-

tal health” items. The third factor loads most highly on “general health perception” items—those with names having the letters ghp in them. The other items describe “physical health”. These designations are based primarily on the wording of the questions, which is summarized in the variable labels.

```
. rotate, varimax
```

```
Factor analysis/correlation      Number of obs   =      9,999
Method: principal factors       Retained factors =        3
Rotation: orthogonal varimax (Kaiser off)  Number of params =      54
```

| Factor | Variance | Difference | Proportion | Cumulative |
|---------|----------|------------|------------|------------|
| Factor1 | 4.20556 | 0.83302 | 0.4358 | 0.4358 |
| Factor2 | 3.37253 | 0.33756 | 0.3495 | 0.7852 |
| Factor3 | 3.03497 | . | 0.3145 | 1.0997 |

LR test: independent vs. saturated: $\chi^2(171) = 1.0e+05$ Prob> $\chi^2 = 0.0000$

Rotated factor loadings (pattern matrix) and unique variances

| Variable | Factor1 | Factor2 | Factor3 | Uniqueness |
|----------|---------|---------|---------|------------|
| ghp31 | -0.2968 | -0.1647 | -0.6567 | 0.4535 |
| pf01 | 0.5872 | 0.0263 | 0.3699 | 0.5177 |
| pf02 | 0.7740 | 0.0848 | 0.2287 | 0.3415 |
| pf03 | 0.7386 | 0.0580 | 0.2654 | 0.3807 |
| pf04 | 0.7484 | 0.0842 | 0.2018 | 0.3921 |
| pf05 | 0.7256 | 0.1063 | 0.1518 | 0.4392 |
| pf06 | 0.5023 | 0.1268 | 0.0730 | 0.7263 |
| rkeep | 0.6023 | 0.2048 | 0.3282 | 0.4876 |
| rkind | 0.6590 | 0.1669 | 0.3597 | 0.4085 |
| sact0 | 0.4187 | 0.3875 | 0.3342 | 0.5628 |
| mha01 | 0.1467 | 0.6859 | 0.1803 | 0.4755 |
| mhp03 | -0.0613 | -0.7375 | -0.1514 | 0.4294 |
| mhd02 | 0.0921 | 0.7893 | 0.1416 | 0.3485 |
| mhp01 | -0.0570 | -0.7671 | -0.1612 | 0.3824 |
| mhc01 | 0.1102 | 0.7124 | 0.1359 | 0.4618 |
| ghp01 | 0.2783 | 0.1977 | 0.6797 | 0.4215 |
| ghp04 | -0.2652 | -0.1908 | -0.7264 | 0.3656 |
| ghp02 | -0.2986 | -0.2116 | -0.7690 | 0.2748 |
| ghp05 | 0.1755 | 0.4756 | 0.4748 | 0.5175 |

Factor rotation matrix

| | Factor1 | Factor2 | Factor3 |
|---------|---------|---------|---------|
| Factor1 | 0.6658 | 0.4796 | 0.5715 |
| Factor2 | 0.5620 | -0.8263 | 0.0387 |
| Factor3 | 0.4908 | 0.2954 | -0.8197 |

With rotation, the structure of the data becomes much clearer. The first rotated factor is physical health, the second is mental health, and the third is general health perception. The a priori designation of the items is confirmed.

After rotation, physical health is the first factor. `rotate` has ordered the factors by explained variance. Still, we warn that the importance of any factor must be gauged against the number of variables that purportedly measure it. Here we included nine variables that measured physical health, five that

measured mental health, and five that measured general health perception. Had we started with only one mental health item, it would have had a high uniqueness, but we would not want to conclude that it was, therefore, largely noise.



□ Technical note

Some people prefer specifying the option `normalize` to apply a Kaiser normalization ([Horst 1965](#)), which places equal weight on all rows of the matrix to be rotated.



▷ Example 9: Oblique oblimin rotation

The literature suggests that physical health and mental health are related. Also, general health perception may be largely a combination of the two. For these reasons, an oblique rotation of a two-factor solution is worth trying. We try the oblique oblimin rotation ([Harman 1976](#)).

```
. factor ghp31-ghp05, factors(2)
(obs=9,999)
```

| | | | |
|-----------------------------|------------------|---|-------|
| Factor analysis/correlation | Number of obs | = | 9,999 |
| Method: principal factors | Retained factors | = | 2 |
| Rotation: (unrotated) | Number of params | = | 37 |

| Factor | Eigenvalue | Difference | Proportion | Cumulative |
|----------|------------|------------|------------|------------|
| Factor1 | 7.27086 | 4.90563 | 0.7534 | 0.7534 |
| Factor2 | 2.36523 | 1.38826 | 0.2451 | 0.9985 |
| Factor3 | 0.97697 | 1.00351 | 0.1012 | 1.0997 |
| Factor4 | -0.02654 | 0.00538 | -0.0027 | 1.0970 |
| Factor5 | -0.03191 | 0.00378 | -0.0033 | 1.0937 |
| Factor6 | -0.03569 | 0.00353 | -0.0037 | 1.0900 |
| Factor7 | -0.03922 | 0.00271 | -0.0041 | 1.0859 |
| Factor8 | -0.04193 | 0.00662 | -0.0043 | 1.0815 |
| Factor9 | -0.04855 | 0.01015 | -0.0050 | 1.0765 |
| Factor10 | -0.05870 | 0.00250 | -0.0061 | 1.0704 |
| Factor11 | -0.06120 | 0.00224 | -0.0063 | 1.0641 |
| Factor12 | -0.06344 | 0.00376 | -0.0066 | 1.0575 |
| Factor13 | -0.06720 | 0.00345 | -0.0070 | 1.0506 |
| Factor14 | -0.07065 | 0.00185 | -0.0073 | 1.0432 |
| Factor15 | -0.07250 | 0.00033 | -0.0075 | 1.0357 |
| Factor16 | -0.07283 | 0.00772 | -0.0075 | 1.0282 |
| Factor17 | -0.08055 | 0.01190 | -0.0083 | 1.0198 |
| Factor18 | -0.09245 | 0.00649 | -0.0096 | 1.0103 |
| Factor19 | -0.09894 | . | -0.0103 | 1.0000 |

LR test: independent vs. saturated: $\chi^2(171) = 1.0e+05$ Prob> $\chi^2 = 0.0000$

Factor loadings (pattern matrix) and unique variances

| Variable | Factor1 | Factor2 | Uniqueness |
|----------|---------|---------|------------|
| ghp31 | -0.6519 | -0.0562 | 0.5718 |
| pf01 | 0.6150 | 0.3226 | 0.5178 |
| pf02 | 0.6867 | 0.3737 | 0.3888 |
| pf03 | 0.6712 | 0.3774 | 0.4070 |
| pf04 | 0.6540 | 0.3588 | 0.4435 |
| pf05 | 0.6209 | 0.3258 | 0.5084 |
| pf06 | 0.4370 | 0.1803 | 0.7765 |
| rkeep | 0.6868 | 0.1820 | 0.4952 |
| rkind | 0.7244 | 0.2464 | 0.4145 |
| sact0 | 0.6556 | -0.0719 | 0.5650 |
| mha01 | 0.5297 | -0.4773 | 0.4916 |
| mhp03 | -0.4810 | 0.5691 | 0.4448 |
| mhd02 | 0.5208 | -0.5949 | 0.3748 |
| mhp01 | -0.4980 | 0.5955 | 0.3974 |
| mhc01 | 0.4927 | -0.5215 | 0.4853 |
| ghp01 | 0.6686 | 0.0194 | 0.5526 |
| ghp04 | -0.6833 | -0.0195 | 0.5327 |
| ghp02 | -0.7398 | -0.0227 | 0.4522 |
| ghp05 | 0.6163 | -0.2760 | 0.5439 |

. rotate, oblimin oblique

| | | |
|--|--------------------|-------|
| Factor analysis/correlation | Number of obs = | 9,999 |
| Method: principal factors | Retained factors = | 2 |
| Rotation: oblique oblimin (Kaiser off) | Number of params = | 37 |

| Factor | Variance | Proportion | Rotated factors are correlated |
|---------|----------|------------|--------------------------------|
| Factor1 | 6.58719 | 0.6826 | |
| Factor2 | 4.65444 | 0.4823 | |

LR test: independent vs. saturated: $\chi^2(171) = 1.0e+05$ Prob> $\chi^2 = 0.0000$

Rotated factor loadings (pattern matrix) and unique variances

| Variable | Factor1 | Factor2 | Uniqueness |
|----------|---------|---------|------------|
| ghp31 | -0.5517 | -0.2051 | 0.5718 |
| pf01 | 0.7179 | -0.0747 | 0.5178 |
| pf02 | 0.8115 | -0.0968 | 0.3888 |
| pf03 | 0.8022 | -0.1068 | 0.4070 |
| pf04 | 0.7750 | -0.0951 | 0.4435 |
| pf05 | 0.7249 | -0.0756 | 0.5084 |
| pf06 | 0.4743 | -0.0044 | 0.7765 |
| rkeep | 0.6712 | 0.0939 | 0.4952 |
| rkind | 0.7478 | 0.0449 | 0.4145 |
| sact0 | 0.4608 | 0.3340 | 0.5650 |
| mha01 | 0.0652 | 0.6869 | 0.4916 |
| mhp03 | 0.0401 | -0.7587 | 0.4448 |
| mhd02 | -0.0280 | 0.8003 | 0.3748 |
| mhp01 | 0.0462 | -0.7918 | 0.3974 |
| mhc01 | 0.0039 | 0.7160 | 0.4853 |
| ghp01 | 0.5378 | 0.2484 | 0.5526 |
| ghp04 | -0.5494 | -0.2541 | 0.5327 |
| ghp02 | -0.5960 | -0.2736 | 0.4522 |
| ghp05 | 0.2805 | 0.5213 | 0.5439 |

Factor rotation matrix

| | Factor1 | Factor2 |
|---------|---------|---------|
| Factor1 | 0.9277 | 0.6831 |
| Factor2 | 0.3733 | -0.7303 |

The first factor is defined predominantly by physical health and the second by mental health. General health perception loads on both, but more on physical health than mental health. To compare the rotated and unrotated solution, looking at both in parallel form is often useful.

```
. estat rotatecompare
Rotation matrix — oblique oblimin (Kaiser off)
```

| Variable | Factor1 | Factor2 |
|----------|---------|---------|
| Factor1 | 0.9277 | 0.6831 |
| Factor2 | 0.3733 | -0.7303 |

Factor loadings

| Variable | Rotated | | Unrotated | |
|----------|---------|---------|-----------|---------|
| | Factor1 | Factor2 | Factor1 | Factor2 |
| ghp31 | -0.5517 | -0.2051 | -0.6519 | -0.0562 |
| pf01 | 0.7179 | -0.0747 | 0.6150 | 0.3226 |
| pf02 | 0.8115 | -0.0968 | 0.6867 | 0.3737 |
| pf03 | 0.8022 | -0.1068 | 0.6712 | 0.3774 |
| pf04 | 0.7750 | -0.0951 | 0.6540 | 0.3588 |
| pf05 | 0.7249 | -0.0756 | 0.6209 | 0.3258 |
| pf06 | 0.4743 | -0.0044 | 0.4370 | 0.1803 |
| rkeep | 0.6712 | 0.0939 | 0.6868 | 0.1820 |
| rkind | 0.7478 | 0.0449 | 0.7244 | 0.2464 |
| sact0 | 0.4608 | 0.3340 | 0.6556 | -0.0719 |
| mha01 | 0.0652 | 0.6869 | 0.5297 | -0.4773 |
| mhp03 | 0.0401 | -0.7587 | -0.4810 | 0.5691 |
| mhd02 | -0.0280 | 0.8003 | 0.5208 | -0.5949 |
| mhp01 | 0.0462 | -0.7918 | -0.4980 | 0.5955 |
| mhc01 | 0.0039 | 0.7160 | 0.4927 | -0.5215 |
| ghp01 | 0.5378 | 0.2484 | 0.6686 | 0.0194 |
| ghp04 | -0.5494 | -0.2541 | -0.6833 | -0.0195 |
| ghp02 | -0.5960 | -0.2736 | -0.7398 | -0.0227 |
| ghp05 | 0.2805 | 0.5213 | 0.6163 | -0.2760 |

Look again at the [factor output](#). The variances of the first and second factor of the unrotated solution are 7.27 and 2.37, respectively. After an orthogonal rotation, the explained variance of $7.27 + 2.37$ is distributed differently over the two factors. For instance, if we typed `rotate, varimax` to obtain an orthogonal varimax rotation, we would see that the first factor has variance 5.75, and the second factor has 3.88—within rounding error $7.27 + 2.37 = 5.75 + 3.88$. The situation after an oblique rotation is different. The variances of the first and second factors are 6.59 and 4.65, which add up to more than in the orthogonal case. In the oblique case, the common factors are correlated and thus “partly explain the same variance”. Therefore, the cumulative proportion of variance explained by the factors is not displayed here.

Most researchers would not be willing to accept a solution in which the common factors are highly correlated.

```
. estat common
Correlation matrix of the oblimin(0) rotated common factors
```

| Factors | Factor1 | Factor2 |
|---------|---------|---------|
| Factor1 | 1 | |
| Factor2 | .3611 | 1 |

The correlation of .36 seems acceptable, so we think that the oblique rotation was a success here.



Factor scores

The `predict` command creates a set of new variables that are estimates of the first k common factors produced by `factor`, `factormat`, or `rotate`. Two types of scoring are available: regression or Thomson scoring and Bartlett scoring.

The number of variables may be less than the number of factors. If so, the first such factors will be used. If the number of variables is greater than the number of factors created or rotated, the unused factors will be filled with missing values.

► Example 10: Predicting scores

Using our automobile data, we wish to develop an index of roominess on the basis of a car's headroom, rear-seat leg room, and trunk space. We begin by extracting the factors of the three variables:

```
. use https://www.stata-press.com/data/r19/autofull
(Automobile models)
. factor headroom rear_seat trunk
(obs=74)
```

```
Factor analysis/correlation      Number of obs   =      74
Method: principal factors       Retained factors =      1
Rotation: (unrotated)          Number of params =      3
```

| Factor | Eigenvalue | Difference | Proportion | Cumulative |
|---------|------------|------------|------------|------------|
| Factor1 | 1.71426 | 1.79327 | 1.1799 | 1.1799 |
| Factor2 | -0.07901 | 0.10329 | -0.0544 | 1.1255 |
| Factor3 | -0.18231 | . | -0.1255 | 1.0000 |

```
LR test: independent vs. saturated:  chi2(3) = 82.93 Prob>chi2 = 0.0000
```

Factor loadings (pattern matrix) and unique variances

| Variable | Factor1 | Uniqueness |
|-----------|---------|------------|
| headroom | 0.7280 | 0.4700 |
| rear_seat | 0.7144 | 0.4897 |
| trunk | 0.8209 | 0.3261 |

All the factor loadings are positive, so we have indeed obtained a “roominess” factor. The `predict` command will now create the one retained factor, which we will call `f1`:

```
. predict f1
(option regression assumed; regression scoring)
Scoring coefficients (method = regression)
```

| Variable | Factor1 |
|-----------|---------|
| headroom | 0.28323 |
| rear_seat | 0.26820 |
| trunk | 0.45964 |

The table with scoring coefficients informs us that the factor is obtained as a weighted sum of standardized versions of `headroom`, `rear_seat`, and `trunk` with weights 0.28, 0.27, and 0.46.

If `factor` had retained more than one factor, typing `predict f1` would still have added only the first factor to our data. Typing `predict f1 f2`, however, would have added the first two factors to our data. `f1` is now our “roominess” index, so we might compare the roominess of domestic and foreign cars:

```
. table foreign, statistic(mean f1) statistic(sd f1)
```

| | Mean | Standard deviation |
|----------|-----------|--------------------|
| Foreign | | |
| Domestic | .2022442 | .9031404 |
| Foreign | -.4780318 | .6106609 |
| Total | 4.51e-09 | .8804116 |

We find that domestic cars are, on average, roomier than foreign cars, at least in our data.

◀

□ Technical note

Are common factors not supposed to be normalized to have mean 0 and standard deviation 1? In our example above, the mean is 4.5×10^{-9} and the standard deviation is 0.88. Why is that?

For the mean, the deviation from zero is due to numerical roundoff, which would diminish dramatically if we had typed `predict double f1` instead. The explanation for the standard deviation of 0.88, on the other hand, is not numerical roundoff. At a theoretical level, the factor is supposed to have standard deviation 1, but the estimation method almost never yields that result unless an exact solution to the factor model is found. This happens for the same reason that, when you regress y on x , you do not get the same equation as if you regress x on y , unless x and y are perfectly collinear.

By the way, if you had two factors, you would expect the correlation between the two factors to be zero because that is how they are theoretically defined. The matrix algebra, however, does not usually work out that way. It is somewhat analogous to the fact that if you regress y on x and the regression assumption that the errors are uncorrelated with the dependent variable is satisfied, then it automatically cannot be satisfied if you regress x on y .

The covariance matrix of the estimated factors is

$$E(\hat{\mathbf{f}}\hat{\mathbf{f}}') = \mathbf{I} - (\mathbf{I} + \mathbf{\Gamma})^{-1}$$

where

$$\mathbf{\Gamma} = \mathbf{\Lambda}'\mathbf{\Psi}^{-1}\mathbf{\Lambda}$$

The columns of $\mathbf{\Lambda}$ are orthogonal to each other, but the inclusion of $\mathbf{\Psi}$ in the middle of the equation destroys that relationship unless all the elements of $\mathbf{\Psi}$ are equal. □

► Example 11: Rescaling the scores

Let’s pretend that we work for the K. E. Watt Company, a fictional industry group that generates statistics on automobiles. Our “roominess” index has mean 0 and standard deviation 0.88, but indexes we present to the public generally have mean 100 and standard deviation 10. First, we wish to rescale our index:

```
. generate roomidx = (f1/.88041161)*10 + 100
. table foreign, statistic(mean roomidx) statistic(sd roomidx) statistic(freq)
> nformat(%9.2f)
```

| | Mean | Standard deviation | Frequency |
|----------|--------|--------------------|-----------|
| Foreign | | | |
| Domestic | 102.30 | 10.26 | 52.00 |
| Foreign | 94.57 | 6.94 | 22.00 |
| Total | 100.00 | 10.00 | 74.00 |

Now when we release our results, we can write, “The K. E. Watt index of roominess shows that domestic cars are, on average, roomier, with an index of 102 versus only 95 for foreign cars.”

Now let’s find the “roomiest” car in our data:

```
. sort roomidx
. list fullname roomidx in 1
```

| | fullname | roomidx |
|-----|---------------|----------|
| 74. | Merc. Marquis | 116.7469 |

We can also write, “K. E. Watt finds that the Mercury Marquis is the roomiest automobile among those surveyed, with a roominess index of 117 versus an average of 100.” ◀

□ Technical note

`predict` provides two methods of scoring: the default regression scoring, which we have used above, and the optional Bartlett method. An artificial example will best illustrate the use and meaning of the methods. We begin by creating a known-to-be-correct factor model in which the true loadings are 0.4, 0.6, and 0.8. The variances of the unique factors are $1 - 0.4^2 = 0.84$, $1 - 0.6^2 = 0.64$, and $1 - 0.8^2 = 0.36$, respectively. We make the sample size large enough so that random fluctuations are not important.

```
. drop _all
. set seed 12345
. set obs 10000
Number of observations (_N) was 0, now 10,000.
. generate ftrue = rnormal()
. generate x1 = .4*ftrue + sqrt(.84)*rnormal()
. generate x2 = .6*ftrue + sqrt(.64)*rnormal()
```

```
. generate x3 = .8*ftrue + sqrt(.36)*rnormal()
. summarize x1 x2 x3
```

| Variable | Obs | Mean | Std. dev. | Min | Max |
|----------|--------|----------|-----------|-----------|----------|
| x1 | 10,000 | .0195519 | 1.011165 | -3.778123 | 4.267452 |
| x2 | 10,000 | .0127835 | 1.001259 | -3.828994 | 4.102375 |
| x3 | 10,000 | .0058335 | 1.002475 | -3.595906 | 3.89754 |

Because we concocted our data, the iterated principal-factor method reproduces the true loadings most faithfully:

```
. factor x1 x2 x3, ipf factors(1)
(obs=10,000)
```

```
Factor analysis/correlation          Number of obs   =    10,000
Method: iterated principal factors    Retained factors =      1
Rotation: (unrotated)                Number of params =      3
```

| Factor | Eigenvalue | Difference | Proportion | Cumulative |
|---------|------------|------------|------------|------------|
| Factor1 | 1.16678 | 1.16662 | 1.0000 | 1.0000 |
| Factor2 | 0.00016 | 0.00036 | 0.0001 | 1.0002 |
| Factor3 | -0.00020 | . | -0.0002 | 1.0000 |

```
LR test: independent vs. saturated:  chi2(3)  = 3887.29 Prob>chi2 = 0.0000
```

Factor loadings (pattern matrix) and unique variances

| Variable | Factor1 | Uniqueness |
|----------|---------|------------|
| x1 | 0.4156 | 0.8273 |
| x2 | 0.6046 | 0.6345 |
| x3 | 0.7928 | 0.3715 |

Let us now compare regression and Bartlett scoring:

```
. predict freg
(option regression assumed; regression scoring)
Scoring coefficients (method = regression)
```

| Variable | Factor1 |
|----------|---------|
| x1 | 0.14449 |
| x2 | 0.27410 |
| x3 | 0.61377 |

```
. predict fbar, bartlett
Scoring coefficients (method = Bartlett)
```

| Variable | Factor1 |
|----------|---------|
| x1 | 0.20285 |
| x2 | 0.38475 |
| x3 | 0.86162 |

Comparing the two scoring vectors, we see that Bartlett scoring yields larger coefficients. The regression scoring method is biased insofar as $E(\text{freg}|\text{ftrue})$ is not ftrue , something we can reveal by regressing freg on ftrue :

```
. regress freg ftrue
```

| Source | SS | df | MS | Number of obs | = | 10,000 |
|----------|------------|-------|------------|---------------|---|----------|
| Model | 5107.57467 | 1 | 5107.57467 | F(1, 9998) | = | 25339.34 |
| Residual | 2015.26671 | 9,998 | .201566984 | Prob > F | = | 0.0000 |
| | | | | R-squared | = | 0.7171 |
| Total | 7122.84138 | 9,999 | .712355374 | Adj R-squared | = | 0.7170 |
| | | | | Root MSE | = | .44896 |

| freg | Coefficient | Std. err. | t | P> t | [95% conf. interval] | |
|-------|-------------|-----------|--------|-------|----------------------|-----------|
| ftrue | .7169557 | .004504 | 159.18 | 0.000 | .708127 | .7257843 |
| _cons | -.0088417 | .00449 | -1.97 | 0.049 | -.0176429 | -.0000404 |

Note the coefficient on ftrue of $0.717 < 1$. The Bartlett scoring method, on the other hand, is unbiased:

```
. regress fbar ftrue
```

| Source | SS | df | MS | Number of obs | = | 10,000 |
|----------|------------|-------|------------|---------------|---|----------|
| Model | 10065.1734 | 1 | 10065.1734 | F(1, 9998) | = | 25339.33 |
| Residual | 3971.35998 | 9,998 | .397215441 | Prob > F | = | 0.0000 |
| | | | | R-squared | = | 0.7171 |
| Total | 14036.5334 | 9,999 | 1.40379372 | Adj R-squared | = | 0.7170 |
| | | | | Root MSE | = | .63025 |

| fbar | Coefficient | Std. err. | t | P> t | [95% conf. interval] | |
|-------|-------------|-----------|--------|-------|----------------------|-----------|
| ftrue | 1.006458 | .0063226 | 159.18 | 0.000 | .9940642 | 1.018851 |
| _cons | -.0124119 | .006303 | -1.97 | 0.049 | -.024767 | -.0000568 |

The zero bias of the Bartlett method comes at the costs of less accuracy, for example, in terms of the mean squared error.

```
. generate dbar = (fbar - ftrue)^2
. generate dreg = (freg - ftrue)^2
. summarize ftrue fbar freg dbar dreg
```

| Variable | Obs | Mean | Std. dev. | Min | Max |
|----------|--------|-----------|-----------|-----------|----------|
| ftrue | 10,000 | .0123322 | .996866 | -4.196032 | 3.815439 |
| fbar | 10,000 | 2.08e-10 | 1.184818 | -3.78561 | 4.550449 |
| freg | 10,000 | -6.44e-10 | .8440115 | -2.696714 | 3.241498 |
| dbar | 10,000 | .3973295 | .5654751 | 1.31e-09 | 7.656609 |
| dreg | 10,000 | .2812835 | .4053233 | 9.68e-10 | 4.814044 |

Neither estimator follows the assumption that the scaled factor has unit variance. The regression estimator has a variance less than 1, and the Bartlett estimator has a variance greater than 1.

The difference between the two scoring methods is not as important as it might seem because the bias in the regression method is only a matter of scaling and shifting.

```
. correlate freg fbar ftrue
(obs=10,000)
```

| | freg | fbar | ftrue |
|-------|--------|--------|--------|
| freg | 1.0000 | | |
| fbar | 1.0000 | 1.0000 | |
| ftrue | 0.8468 | 0.8468 | 1.0000 |

Therefore, the choice of which scoring method we apply is largely immaterial.



Stored results

Let p be the number of variables and f , the number of factors.

`predict`, in addition to generating variables, also stores the following in `r()`:

Macros

`r(method)` regression or Bartlett

Matrices

`r(scoef)` $p \times f$ matrix of scoring coefficients

`estat anti` stores the following in `r()`:

Matrices

`r(acov)` $p \times p$ anti-image covariance matrix

`r(acorr)` $p \times p$ anti-image correlation matrix

`estat common` stores the following in `r()`:

Matrices

`r(Phi)` $f \times f$ correlation matrix of common factors

`estat factors` stores the following in `r()`:

Matrices

`r(stats)` $k \times 5$ matrix with log likelihood, degrees of freedom, AIC, and BIC for models with 1 to k factors estimated via maximum likelihood

`estat kmo` stores the following in `r()`:

Scalars

`r(kmo)` the Kaiser–Meyer–Olkin measure of sampling adequacy

Matrices

`r(kmow)` column vector of KMO measures for each variable

`estat residuals` stores the following in `r()`:

Matrices

`r(fit)` fitted matrix for the correlations, $\widehat{C} = \widehat{\Lambda}\widehat{\Phi}\widehat{\Lambda}' + \widehat{\Psi}$

`r(res)` raw residual matrix $C - \widehat{C}$

`r(SR)` standardized residuals (`sresiduals` option only)

`estat smc` stores the following in `r()`:

Matrices

`r(smc)` vector of squared multiple correlations of variables with all other variables

`estat structure` stores the following in `r()`:

Matrices

`r(st)` $p \times f$ matrix of correlations between variables and common factors

See [R] [estat summarize](#) for the [stored results](#) of `estat summarize`.

`rotate` after `factor` and `factormat` add to the existing `e()`:

Scalars

`e(r_f)` number of factors in rotated solution

`e(r_fmin)` rotation criterion value

Macros

`e(r_class)` orthogonal or oblique

`e(r_criterion)` rotation criterion

`e(r_ctitle)` title for rotation

`e(r_normalization)` kaiser or none

Matrices

`e(r_L)` rotated loadings

`e(r_T)` rotation

`e(r_Phi)` correlations between common factors

`e(r_Ev)` explained variance by common factors

The factors in the rotated solution are in decreasing order of `e(r_Ev)`.

Methods and formulas

Methods and formulas are presented under the following headings:

[estat](#)
[rotate](#)
[predict](#)

estat

See [Methods and formulas](#) of [MV] [pca postestimation](#) for the formulas for `estat anti`, `estat kmo`, and `estat smc`.

`estat residuals` computes the standardized residuals \tilde{r}_{ij} as

$$\tilde{r}_{ij} = \frac{\sqrt{N}(r_{ij} - f_{ij})}{\sqrt{f_{ij}^2 + f_{ii}f_{jj}}}$$

suggested by [Jöreskog and Sörbom \(1986\)](#), where N is the number of observations, r_{ij} is the observed correlation of variables i and j , and f_{ij} is the fitted correlation of variables i and j . Also see [Bollen \(1989\)](#). Caution is warranted in interpretation of these residuals; see [Jöreskog and Sörbom \(1988\)](#).

`estat structure` computes the correlations of the variables and the common factors as $\Lambda\Phi$.

rotate

See *Methods and formulas* of [MV] **rotatemat** for the details of rotation.

The correlation of common factors after rotation is $\mathbf{T}'\mathbf{T}$, where \mathbf{T} is the factor rotation matrix, satisfying $\mathbf{L}_{\text{rotated}} = \mathbf{L}_{\text{unrotated}}(\mathbf{T}')^{-1}$

predict

The formula for regression scoring (Thomson 1951) in the orthogonal case is

$$\hat{\mathbf{f}} = \mathbf{\Lambda}'\mathbf{\Sigma}^{-1}\mathbf{x}$$

where $\mathbf{\Lambda}$ is the unrotated or orthogonally rotated loading matrix. For oblique rotation, the regression scoring is defined as

$$\hat{\mathbf{f}} = \mathbf{\Phi}\mathbf{\Lambda}'\mathbf{\Sigma}^{-1}\mathbf{x}$$

where $\mathbf{\Phi}$ is the correlation matrix of the common factors.

The formula for Bartlett scoring (Bartlett 1937, 1938) is

$$\mathbf{\Gamma}^{-1}\mathbf{\Lambda}'\mathbf{\Psi}^{-1}\mathbf{x}$$

where

$$\mathbf{\Gamma} = \mathbf{\Lambda}'\mathbf{\Psi}^{-1}\mathbf{\Lambda}$$

See Harman (1976) and Lawley and Maxwell (1971).

References

- Akaike, H. 1987. Factor analysis and AIC. *Psychometrika* 52: 317–332. <https://doi.org/10.1007/BF02294359>.
- Baldwin, S. 2019. *Psychological Statistics and Psychometrics Using Stata*. College Station, TX: Stata Press.
- Bartlett, M. S. 1937. The statistical conception of mental factors. *British Journal of Psychology* 28: 97–104. <https://doi.org/10.1111/j.2044-8295.1937.tb00863.x>.
- . 1938. Methods of estimating mental factors. *Nature* 141: 609–610.
- Bollen, K. A. 1989. *Structural Equations with Latent Variables*. New York: Wiley. <https://doi.org/10.1002/9781118619179>.
- Cattell, R. B. 1966. The scree test for the number of factors. *Multivariate Behavioral Research* 1: 245–276. https://doi.org/10.1207/s15327906mbr0102_10.
- Harman, H. H. 1976. *Modern Factor Analysis*. 3rd ed. Chicago: University of Chicago Press.
- Horst, P. 1965. *Factor Analysis of Data Matrices*. New York: Holt, Rinehart and Winston.
- Jöreskog, K. G., and D. Sörbom. 1986. *Lisrel VI: Analysis of linear structural relationships by the method of maximum likelihood*. Mooresville, IN: Scientific Software.
- . 1988. *PRELIS: A program for multivariate data screening and data summarization. A preprocessor for LISREL*. 2nd ed. Mooresville, IN: Scientific Software.
- Kaiser, H. F. 1958. The varimax criterion for analytic rotation in factor analysis. *Psychometrika* 23: 187–200. <https://doi.org/10.1007/BF02289233>.
- . 1974. An index of factor simplicity. *Psychometrika* 39: 31–36.
- Lawley, D. N., and A. E. Maxwell. 1971. *Factor Analysis as a Statistical Method*. 2nd ed. London: Butterworths.
- Schwarz, G. 1978. Estimating the dimension of a model. *Annals of Statistics* 6: 461–464. <https://doi.org/10.1214/aos/1176344136>.

Tarlov, A. R., J. E. Ware, Jr., S. Greenfield, E. C. Nelson, E. Perrin, and M. Zubkoff. 1989. The medical outcomes study. An application of methods for monitoring the results of medical care. *Journal of the American Medical Association* 262: 925–930. <https://doi.org/10.1001/jama.1989.03430070073033>.

Thomson, G. H. 1951. *The Factorial Analysis of Human Ability*. London: University of London Press.

Also see *References* in [MV] **factor**.

Also see

[MV] **factor** — Factor analysis

[MV] **rotate** — Orthogonal and oblique rotations after factor and pca

[MV] **scoreplot** — Score and loading plots

[MV] **screeplot** — Scree plot of eigenvalues

[U] **20 Estimation and postestimation commands**

Stata, Stata Press, and Mata are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow and NetCourseNow are trademarks of StataCorp LLC. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2025 StataCorp LLC, College Station, TX, USA. All rights reserved.

For suggested citations, see the FAQ on [citing Stata documentation](#).

