Title

stata.com

pca postestimation — Postestimation tools for pca and pcamat

Postestimation commands

The following postestimation commands are of special interest after `pca` and `pcamat`:

Command	Description
`estat anti`	anti-image correlation and covariance matrices
`estat kmo`	Kaiser–Meyer–Olkin measure of sampling adequacy
`estat loadings`	component-loading matrix in one of several normalizations
`estat residuals`	matrix of correlation or covariance residuals
`estat rotatecompare`	compare rotated and unrotated components
`estat smc`	squared multiple correlations between each variable and the rest
* `estat summarize`	display summary statistics over the estimation sample
`loadingplot`	plot component loadings
`rotate`	rotate component loadings
`scoreplot`	plot score variables
`screeplot`	plot eigenvalues

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

The following standard postestimation commands are also available:

Command	Description
† `estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
* `lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
* `nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of coefficients
`predict`	score variables, predictions, and residuals
* `predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
* `test`	Wald tests of simple and composite linear hypotheses
* `testnl`	Wald tests of nonlinear hypotheses

† `estat vce` is available after `pca` and `pcamat` with the `vce(normal)` option.

* `lincom`, `nlcom`, `predictnl`, `test`, and `testnl` are available only after `pca` with the `vce(normal)` option.

predict

Description for predict

`predict` creates new variables containing predictions such as scores, fitted values, raw residuals, and residual sums of squares.

Menu for predict

Statistics > Postestimation

Syntax for predict

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

statistic	# of vars.	Description (k = # of orig. vars.; f = # of components)
Main		
`score`	1, ..., f	scores based on the components; the default
`fit`	k	fitted values using the retained components
`residual`	k	raw residuals from the fit using the retained components
`q`	1	residual sums of squares

options	Description
Main	
`norotated`	use unrotated results, even when rotated results are available
`center`	base scores on centered variables
`notable`	suppress table of scoring coefficients
`format(`*%fmt*`)`	format for displaying the scoring coefficients

Options for predict

Note on `pcamat`: `predict` requires that variables with the correct names be available in memory. Apart from centered scores, `means()` should have been specified with `pcamat`. If you used `pcamat` because you have access only to the correlation or covariance matrix, you cannot use `predict`.

Main

`score` calculates the scores for components 1, ..., #, where # is the number of variables in *newvarlist*.

`fit` calculates the fitted values, using the retained components, for each variable. The number of variables in *newvarlist* should equal the number of variables in the *varlist* of `pca`; see [MV] **pca**.

`residual` calculates for each variable the raw residuals (residual = observed − fitted), with the fitted values computed using the retained components.

`q` calculates the Rao statistics (that is, the sums of squares of the omitted components) weighted by the respective eigenvalues. This equals the residual sums of squares between the original variables and the fitted values.

`norotated` uses unrotated results, even when rotated results are available.

`center` bases scores on centered variables. This option is relevant only for a PCA of a covariance matrix, in which the scores are based on uncentered variables by default. Scores for a PCA of a correlation matrix are always based on the standardized variables.

`notable` suppresses the table of scoring coefficients.

`format(%`*fmt*`)` specifies the display format for scoring coefficients. The default is `format(%8.4f)`.

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 of all pairs of variables, holding all other variables constant.

`estat kmo` displays the Kaiser–Meyer–Olkin (KMO) measure of sampling adequacy. KMO takes values between 0 and 1, with small values indicating that overall the variables have too little in common to warrant a PCA. 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 loadings` displays the component-loading matrix in one of several normalizations of the columns (eigenvectors).

`estat residuals` displays the difference between the observed correlation or covariance matrix and the fitted (reproduced) matrix using the retained factors.

`estat rotatecompare` displays the unrotated (principal) components next to the most recent rotated components.

`estat smc` displays the squared multiple correlations between each variable and all other variables. SMC is a theoretical lower bound for communality and thus an upper bound for the unexplained variance.

`estat summarize` displays summary statistics of the variables in the principal component analysis over the estimation sample. This subcommand is not available after `pcamat`.

Menu for estat

Statistics > Postestimation

Syntax for estat

Display the anti-image correlation and covariance matrices

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

Display the Kaiser–Meyer–Olkin measure of sampling adequacy

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

Display the component-loading matrix

```
estat loadings [, cnorm(unit|eigen|inveigen) format(%fmt)]
```

Display the differences in matrices

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

Display the unrotated and rotated components

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

Display the squared multiple correlations

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

Display the summary statistics

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

Options for estat

`nocorr`, an option used with `estat anti`, suppresses the display of the anti-image correlation matrix, that is, minus the partial correlation matrix of all pairs of variables, holding constant all other variables.

`nocov`, an option used with `estat anti`, suppresses the display of the anti-image covariance matrix, that is, minus the partial covariance matrix of all pairs of variables, holding constant all other variables.

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

`novar`, an option used with `estat kmo`, suppresses the Kaiser–Meyer–Olkin measures of sampling adequacy for the variables in the principal component analysis, displaying the overall KMO measure only.

`cnorm(unit|eigen|inveigen)`, an option used with `estat loadings`, selects the normalization of the eigenvectors, the columns of the principal-component loading matrix. The following normalizations are available

`unit`	ssq(column) = 1; the default
`eigen`	ssq(column) = eigenvalue
`inveigen`	ssq(column) = 1/eigenvalue

with ssq(column) being the sum of squares of the elements in a column and eigenvalue, the eigenvalue associated with the column (eigenvector).

`obs`, an option used with `estat residuals`, displays the observed correlation or covariance matrix for which the PCA was performed.

`fitted`, an option used with `estat residuals`, displays the fitted (reconstructed) correlation or covariance matrix based on the retained components.

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

Remarks and examples

stata.com

After computing the principal components and the associated eigenvalues, you have more issues to resolve. How many components do you want to retain? How well is the correlation or covariance matrix approximated by the retained components? How can you interpret the principal components? Is it possible to improve the interpretability by rotating the retained principal components? And, when these issues have been settled, the component scores are probably needed for later research.

The rest of this entry describes the specific tools available for these purposes.

Remarks are presented under the following headings:

In addition to these specific postestimation tools, general tools are available as well. `pca` is an estimation command, so it is possible to manage a series of PCA analyses with the `estimates` command; see [R] **estimates**. If you have specified the `vce(normal)` option, `pca` has stored the coefficients `e(b)` and the associated variance–covariance matrix `e(V)`, and you can use standard Stata commands to test hypotheses about the principal components and eigenvalues ("confirmatory principal component analysis"), for instance, with the `test`, `lincom`, and `testnl` commands. We caution you to test only hypotheses that do not violate the assumptions of the theory underlying the derivation of the covariance matrix. In particular, all eigenvalues are assumed to be different and strictly positive. Thus it makes no sense to use `test` to test the hypothesis that the smallest four eigenvalues are equal (let alone that they are equal to zero.)

Postestimation statistics

`pca` displays the principal components in unit normalization; the sum of squares of the principal loadings equals 1. This parallels the standard conventions in mathematics concerning eigenvectors. Some texts and some software use a different normalization. Some texts multiply the eigenvectors by the square root of the eigenvalues. In this normalization, the sum of the squared loadings equals the variance explained by that component. `estat loadings` can display the loadings in this normalization.

```
. use http://www.stata-press.com/data/r15/audiometric
(Audiometric measures)
. pca l* r*, comp(4)
  (output omitted)
```

```
. estat loadings, cnorm(eigen)
Principal component loadings (unrotated)
    component normalization: sum of squares(column) = eigenvalue
```

	Comp1	Comp2	Comp3	Comp4
lft500	.795	-.4032	.1562	-.2239
lft1000	.8345	-.2868	-.05132	-.3291
lft2000	.7262	.3035	-.4645	-.193
lft4000	.5567	.6032	.4242	-.1101
rght500	.6804	-.4911	.2561	.3331
rght1000	.8155	-.2948	-.0285	.2544
rght2000	.6175	.4033	-.5559	.2674
rght4000	.5039	.6533	.4209	.1087

How close the retained principal components approximate the correlation matrix can be seen from the fitted (reconstructed) correlation matrix and from the residuals, that is, the difference between the observed and fitted correlations.

```
. estat residual, fit format(%7.3f)
Fitted correlation matrix
```

Variable	lft500	lft1000	lft2000	lft4000	rght500	rg~1000
lft500	0.869					
lft1000	0.845	0.890				
lft2000	0.426	0.606	0.872			
lft4000	0.290	0.306	0.412	0.866		
rght500	0.704	0.586	0.162	0.155	0.881	
rght1000	0.706	0.683	0.467	0.236	0.777	0.818
rght2000	0.182	0.340	0.778	0.322	0.169	0.469
rght4000	0.179	0.176	0.348	0.841	0.166	0.234

Variable	rg~2000	rg~4000
rght2000	0.925	
rght4000	0.370	0.870

```
Residual correlation matrix
```

Variable	lft500	lft1000	lft2000	lft4000	rght500	rg~1000
lft500	0.131					
lft1000	-0.067	0.110				
lft2000	-0.024	-0.070	0.128			
lft4000	-0.035	-0.031	0.013	0.134		
rght500	-0.008	-0.034	0.077	0.024	0.119	
rght1000	-0.064	0.024	-0.021	0.027	-0.114	0.182
rght2000	0.056	0.020	-0.076	-0.005	-0.010	-0.054
rght4000	0.025	0.041	-0.022	-0.131	-0.034	-0.014

Variable	rg~2000	rg~4000
rght2000	0.075	
rght4000	0.005	0.130

All off diagonal residuals are small, except perhaps the two measurements at the highest frequency.

`estat` also provides some of the standard methods for studying correlation matrices to assess whether the variables have strong linear relations with each other. In a sense, these methods could be seen as preestimation rather than as postestimation methods. The first method is the inspection of the squared multiple correlation (the regression R^2) of each variable on all other variables.

```
. estat smc
Squared multiple correlations of variables with all other variables
```

Variable	smc
lft500	0.7113
lft1000	0.7167
lft2000	0.6229
lft4000	0.5597
rght500	0.5893
rght1000	0.6441
rght2000	0.5611
rght4000	0.5409

The SMC measures help identify variables that cannot be explained well from the other variables. For such variables, you should reevaluate whether they should be included in the analysis. In our examples, none of the SMCs are so small as to warrant exclusion. Two other statistics are offered. First, we can inspect the anti-image correlation and covariance matrices, that is, the negative of correlations (covariances) of the variables partialing out all other variables. If many of these correlations or covariances are "high", the relationships between some of the variables have little to do with the other variables, indicating that it will not be possible to obtain a low-dimensional reduction of the data.

```
. estat anti, nocov format(%7.3f)
Anti-image correlation coefficients ── partialing out all other variables
```

Variable	lft500	lft1000	lft2000	lft4000	rght500	rg~1000
lft500	1.000					
lft1000	-0.561	1.000				
lft2000	-0.051	-0.267	1.000			
lft4000	-0.014	0.026	-0.285	1.000		
rght500	-0.466	0.131	0.064	-0.017	1.000	
rght1000	0.023	-0.389	0.043	-0.042	-0.441	1.000
rght2000	0.085	0.068	-0.617	0.161	0.067	-0.248
rght4000	-0.047	-0.002	0.150	-0.675	0.019	0.023

Variable	rg~2000	rg~4000
rght2000	1.000	
rght4000	-0.266	1.000

The Kaiser–Meyer–Olkin measure of sampling adequacy compares the correlations and the partial correlations between variables. If the partial correlations are relatively high compared to the correlations, the KMO measure is small, and a low-dimensional representation of the data is not possible.

```
. estat kmo
Kaiser-Meyer-Olkin measure of sampling adequacy
```

Variable	kmo
lft500	0.7701
lft1000	0.7767
lft2000	0.7242
lft4000	0.6449
rght500	0.7562
rght1000	0.8168
rght2000	0.6673
rght4000	0.6214
Overall	0.7328

Using the Kaiser (1974) characterization of KMO values,

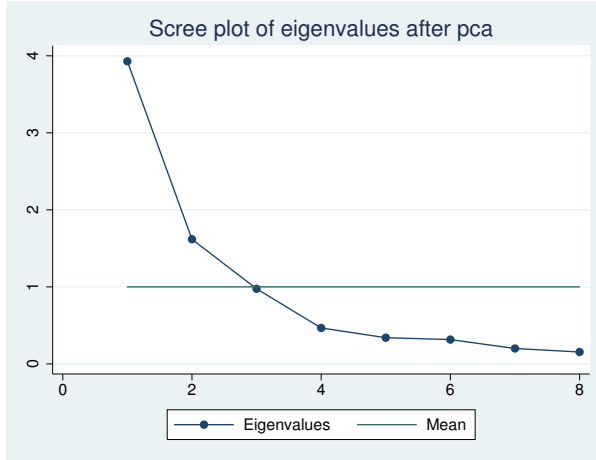
0.00 to 0.49	unacceptable
0.50 to 0.59	miserable
0.60 to 0.69	mediocre
0.70 to 0.79	middling
0.80 to 0.89	meritorious
0.90 to 1.00	marvelous

we declare our KMO value, 0.73, middling.

Plots of eigenvalues, component loadings, and scores

After computing the principal components, we probably wish to determine how many components to keep. In factor analysis the question of the "true" number of factors is a complicated one. With PCA, it is a little more straightforward. We may set a percentage of variance we wish to account for, say, 90%, and retain just enough components to account for at least that much of the variance. Usually you will want to weigh the costs associated with using more components in later analyses against the benefits of the extra variance they account for. The relative magnitudes of the eigenvalues indicate the amount of variance they account for. A useful tool for visualizing the eigenvalues relative to one another, so that you can decide the number of components to retain, is the scree plot proposed by Cattell (1966); see [MV] **screeplot**.

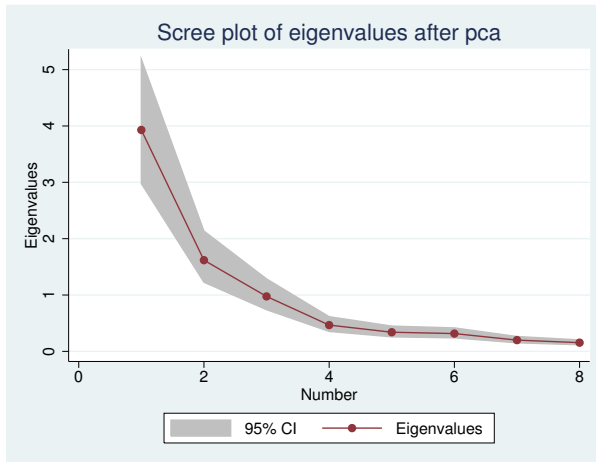
```
. screeplot, mean
```



Because we are analyzing a correlation matrix, the mean eigenvalue is 1. We wish to retain the components associated with the high part of the scree plot and drop the components associated with the lower flat part of the scree plot. The boundary between high and low is not clear here, but we would choose two or three components, although the fourth component had the nice interpretation of the left versus the right ear; see [MV] **pca**.

A problem in interpreting the scree plot is that no guidance is given with respect to its stability under sampling. How different could the plot be with different samples? The approximate variance of an eigenvalue $\widehat{\lambda}$ of a covariance matrix for multivariate normal distributed data is $2\lambda^2/n$. From this we can derive confidence intervals for the eigenvalues. These scree plot confidence intervals aid in the selection of important components.

```
. screeplot, ci
(caution is advised in interpreting an asymptotic theory-based confidence
 interval of eigenvalues of a correlation matrix)
```

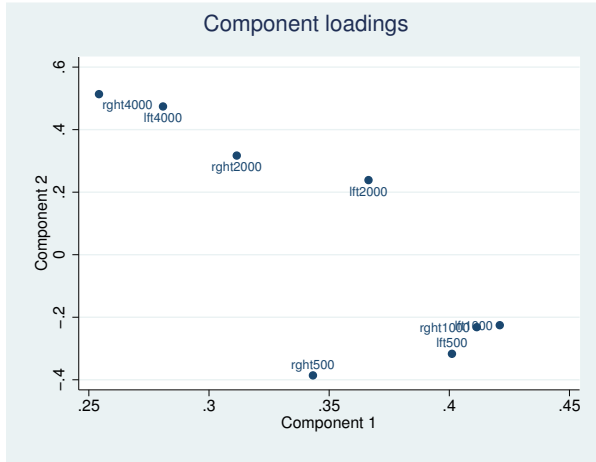


Despite our appreciation of the underlying interpretability of the fourth component, the evidence still points to retaining two or three principal components.

Plotting the components is sometimes useful in interpreting a PCA. We may look at the components from the perspective of the columns (variables) or the rows (observations). The associated plots are produced by the commands `loadingplot` (variables) and `scoreplot` (observations).

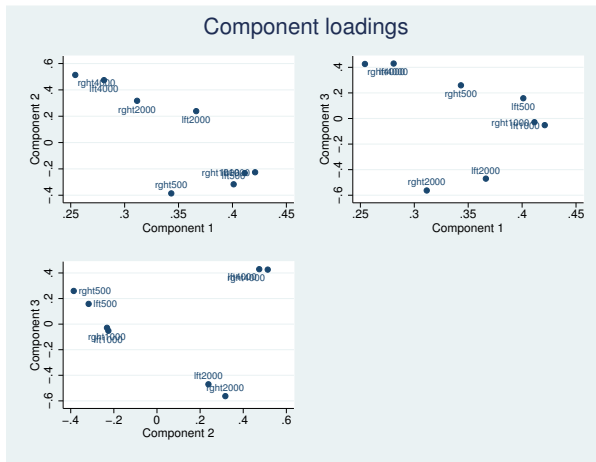
By default, the first two components are used to produce the loading plot.

```
. loadingplot
```



You may request more components, in which case each possible pair of requested components will be graphed. You can choose between a matrix or combined graph layout for the multiple graphs. Here we show the combined layout.

```
. loadingplot, comp(3) combined
```



Score plots approach the display of principal components from the perspective of the observations. `scoreplot` and `loadingplot` have most of their options in common; see [MV] **scoreplot**. Unlike `loadingplot`, which automatically uses the variable names as marker labels, with `scoreplot` you use the `mlabel()` graph option to provide meaningful marker labels. Score plots are especially helpful if the observations are well-known objects, such as countries, firms, or brands. The score plot may

help you visualize the principal components with your background knowledge of these objects. Score plots are sometimes useful for detecting outliers; see Jackson (2003).

❑ Technical note

In [MV] **pca**, we noted that PCA may also be interpreted as fixed-effects factor analysis; in that interpretation, the selection of the number of components to be retained is of comparable complexity as in factor analysis.

❑

Rotating the components

Rotating principal components is a disputed issue and one in which reasonable people may disagree. `pca` computes the principal components. Rotating the solution destroys some of the properties of principal components. In particular, the first rotated component no longer has maximal variance, the second rotated component no longer has maximal variance among those linear combinations uncorrelated to the first component, etc. If preserving the maximal variance property is very important to your interpretations, do not rotate.

On the other hand, when we rotate, say, the leading three principal components, the total variance explained by the three rotated components is equal to the variance explained by the three principal components. If you applied an orthogonal rotation, the rotated components are still uncorrelated. The only thing that has changed is that the explanation is distributed differently among the three rotated components. If the rotated components have a clearer interpretation, you may actually prefer to use them in your subsequent work.

After `pca`, a wide variety of rotations are available; see [MV] **rotate**. The default method of rotation is varimax, rotating the principal components to maximize the sum over the columns of the within-column variances.

```
. rotate
Principal components/correlation                  Number of obs    =        100
                                                  Number of comp.  =          4
                                                  Trace            =          8
    Rotation: orthogonal varimax (Kaiser off)     Rho              =     0.8737
```

Component	Variance	Difference	Proportion	Cumulative
Comp1	2.11361	.400444	0.2642	0.2642
Comp2	1.71316	.118053	0.2141	0.4783
Comp3	1.59511	.0275517	0.1994	0.6777
Comp4	1.56756	.	0.1959	0.8737

Rotated components

Variable	Comp1	Comp2	Comp3	Comp4	Unexplained
lft500	0.5756	0.0265	-0.1733	0.1781	.1308
lft1000	0.6789	-0.0289	-0.0227	-0.0223	.1105
lft2000	0.3933	0.0213	0.5119	-0.2737	.1275
lft4000	0.1231	0.6987	-0.0547	-0.0885	.1342
rght500	-0.0005	0.0158	-0.0380	0.7551	.1194
rght1000	0.0948	-0.0248	0.2289	0.5481	.1825
rght2000	-0.1173	-0.0021	0.8047	0.0795	.07537
rght4000	-0.1232	0.7134	0.0550	0.0899	.1303

Component rotation matrix

	Comp1	Comp2	Comp3	Comp4
Comp1	0.6663	0.3784	0.4390	0.4692
Comp2	-0.3055	0.6998	0.4012	-0.5059
Comp3	-0.0657	0.6059	-0.7365	0.2936
Comp4	-0.6770	-0.0022	0.3224	0.6616

`rotate` now labels one of the columns of the first table as "Variance" instead of "Eigenvalue"; the rotated components have been ordered in decreasing order of variance. The variance explained by the four rotated components equals 87.37%, which is identical to the explained variance by the four leading principal components. But whereas the principal components have rather dispersed eigenvalues, the four rotated components all explain about the same fraction of the variance.

You may also choose to rotate only a few of the retained principal components. In contrast to most methods of factor analysis, the principal components are not affected by the number of retained components. However, the first two rotated components are different if you are rotating all four components or only the leading two or three principal components.

```
. rotate, comp(3)
Principal components/correlation                  Number of obs    =        100
                                                  Number of comp.  =          4
                                                  Trace            =          8
    Rotation: orthogonal varimax (Kaiser off)     Rho              =     0.8737
```

Component	Variance	Difference	Proportion	Cumulative
Comp1	2.99422	1.16842	0.3743	0.3743
Comp2	1.8258	.123163	0.2282	0.6025
Comp3	1.70264	1.23585	0.2128	0.8153
Comp4	.466782	.	0.0583	0.8737

Rotated components

Variable	Comp1	Comp2	Comp3	Comp4	Unexplained
lft500	0.5326	-0.0457	0.0246	-0.3278	.1308
lft1000	0.4512	0.1618	-0.0320	-0.4816	.1105
lft2000	0.0484	0.6401	0.0174	-0.2824	.1275
lft4000	0.0247	0.0011	0.6983	-0.1611	.1342
rght500	0.5490	-0.1799	0.0163	0.4876	.1194
rght1000	0.4521	0.1368	-0.0259	0.3723	.1825
rght2000	-0.0596	0.7148	-0.0047	0.3914	.07537
rght4000	-0.0200	0.0059	0.7138	0.1591	.1303

Component rotation matrix

	Comp1	Comp2	Comp3	Comp4
Comp1	0.7790	0.5033	0.3738	0.0000
Comp2	-0.5932	0.3987	0.6994	0.0000
Comp3	0.2030	-0.7666	0.6092	-0.0000
Comp4	0.0000	-0.0000	-0.0000	1.0000

The three-component varimax-rotated solution differs from the leading three components from the four component varimax-rotated solution. The fourth component is not affected by a rotation among the leading three component—it is still the fourth principal component.

So, how interpretable are rotated components? We believe that for this example the original components had a much clearer interpretation than the rotated components. Notice how the clear symmetry in the treatment of left and right ears has been broken.

To add further to an already controversial method, we may use oblique rotation methods. An example is the oblique oblimin method.

```
. rotate, oblimin oblique
Principal components/correlation                  Number of obs    =        100
                                                  Number of comp.  =          4
                                                  Trace            =          8
    Rotation: oblique oblimin (Kaiser off)        Rho              =     0.8737
```

Component	Variance	Proportion	Rotated comp. are correlated
Comp1	2.21066	0.2763	
Comp2	1.71164	0.2140	
Comp3	1.69708	0.2121	
Comp4	1.62592	0.2032	

Rotated components

Variable	Comp1	Comp2	Comp3	Comp4	Unexplained
lft500	0.5834	0.0259	0.1994	-0.1649	.1308
lft1000	0.6797	-0.0292	0.0055	-0.0157	.1105
lft2000	0.3840	0.0216	-0.2489	0.5127	.1275
lft4000	0.1199	0.6988	-0.0857	-0.0545	.1342
rght500	0.0261	0.0146	0.7561	-0.0283	.1194
rght1000	0.1140	-0.0257	0.5575	0.2370	.1825
rght2000	-0.1158	-0.0022	0.0892	0.8048	.07537
rght4000	-0.1209	0.7134	0.0848	0.0549	.1303

Component rotation matrix

	Comp1	Comp2	Comp3	Comp4
Comp1	0.6836	0.3773	0.5053	0.4523
Comp2	-0.3250	0.7008	-0.5137	0.3916
Comp3	-0.0550	0.6054	0.2774	-0.7337
Comp4	-0.6557	-0.0029	0.6408	0.3238

The oblique rotation methods do not change the variance that is unexplained by the components. But this time, the rotated components are no longer uncorrelated. This makes measuring the importance of the rotated components more ambiguous, a problem that is similar to ambiguities in interpreting importance of correlated independent variables. In this oblique case, the sum of the variances of the rotated components equals 90.6% $(0.2763 + 0.2140 + 0.2121 + 0.2032)$ of the total variance. This is larger than the 87.37% of variance explained by the four principal components. The oblique rotated components partly explain the same variance, and this shared variance is entering multiple times into the total.

How rotate interacts with pca

rotate stores the rotated component loadings and associated statistics in e(), the estimation storage area, along with the regular pca estimation results. Replaying pca will display the rotated results again.

Other postestimation statistics also use the rotated results whenever this is meaningful. For instance, loadingplot would display the rotated loadings. These postestimation commands have an option norotated that specifies that the unrotated results, that is, the principal components, be used. Thus

```
. pca, norotated
  (output omitted)
```

displays the standard pca output for the unrotated (principal) solution, and

```
. loadingplot, norotated
  (output omitted)
```

produces the loading plot for the unrotated (principal) solution.

If you execute rotate again, the new rotate results are stored with the pca estimation, replacing the previous rotate results. Thus pca knows about at most one rotation.

To compare rotated and unrotated results, it is of course possible to replay the rotated results (pca) and unrotated results (pca, norotate) consecutively. You would especially seek to compare the loadings. Such a comparison is easier if the loadings are displayed in parallel. This feature is provided with the estat command rotatecompare.

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

Variable	Comp1	Comp2	Comp3	Comp4
Comp1	0.6836	0.3773	0.5053	0.4523
Comp2	-0.3250	0.7008	-0.5137	0.3916
Comp3	-0.0550	0.6054	0.2774	-0.7337
Comp4	-0.6557	-0.0029	0.6408	0.3238

Rotated component loadings

Variable	Comp1	Comp2	Comp3	Comp4
lft500	0.5834	0.0259	0.1994	-0.1649
lft1000	0.6797	-0.0292	0.0055	-0.0157
lft2000	0.3840	0.0216	-0.2489	0.5127
lft4000	0.1199	0.6988	-0.0857	-0.0545
rght500	0.0261	0.0146	0.7561	-0.0283
rght1000	0.1140	-0.0257	0.5575	0.2370
rght2000	-0.1158	-0.0022	0.0892	0.8048
rght4000	-0.1209	0.7134	0.0848	0.0549

```
Unrotated component loadings
```

Variable	Comp1	Comp2	Comp3	Comp4
lft500	0.4011	-0.3170	0.1582	-0.3278
lft1000	0.4210	-0.2255	-0.0520	-0.4816
lft2000	0.3664	0.2386	-0.4703	-0.2824
lft4000	0.2809	0.4742	0.4295	-0.1611
rght500	0.3433	-0.3860	0.2593	0.4876
rght1000	0.4114	-0.2318	-0.0289	0.3723
rght2000	0.3115	0.3171	-0.5629	0.3914
rght4000	0.2542	0.5135	0.4262	0.1591

Finally, sometimes you may want to remove rotation results permanently; for example, you decide to continue with the unrotated (principal) solution. Because all postestimation commands operate on the rotated solution by default, you would have to add the option `norotated` over and over again. Instead, you can remove the rotated solution with the command

```
. rotate, clear
```

❑ Technical note

`pca` results may be stored and restored with `estimates`, just like other estimation results. If you have stored PCA estimation results without rotated results, and later `rotate` the solution, the rotated results are not automatically stored as well. The `pca` would need to be stored again.

❑

Predicting the component scores

After deciding on the number of components and, possibly, the rotation of the components, you may want to estimate the component scores for all respondents. To estimate only the first component scores, which here is called `pc1`:

```
. predict pc1
(score assumed)
(3 components skipped)
Scoring coefficients
    sum of squares(column-loading) = 1
```

Variable	Comp1	Comp2	Comp3	Comp4
lft500	0.4011	-0.3170	0.1582	-0.3278
lft1000	0.4210	-0.2255	-0.0520	-0.4816
lft2000	0.3664	0.2386	-0.4703	-0.2824
lft4000	0.2809	0.4742	0.4295	-0.1611
rght500	0.3433	-0.3860	0.2593	0.4876
rght1000	0.4114	-0.2318	-0.0289	0.3723
rght2000	0.3115	0.3171	-0.5629	0.3914
rght4000	0.2542	0.5135	0.4262	0.1591

The table is informing you that `pc1` could be obtained as a weighted sum of standardized variables,

```
. egen std_lft500   = std(lft500)
. egen std_lft1000  = std(lft1000)
. egen std_rght4000 = std(rght4000)
. gen pc1 = 0.4011*std_lft500 + 0.4210*std_lft1000 + ... + 0.2542*std_rght4000
```

(egen's std() function converts a variable to its standardized form (mean 0, variance 1); see [D] **egen**.) The principal-component scores are in standardized units after a PCA of a correlation matrix and in the original units after a PCA of a covariance matrix.

It is possible to predict other statistics as well. For instance, the fitted values of the eight variables by the first four principal components are obtained as

```
. predict f_1-f_8, fit
```

The predicted values are in the units of the original variables, with the means substituted back in. If we had retained all eight components, the fitted values would have been identical to the observations.

❑ Technical note

The fitted values are meaningful in the interpretation of PCA as rank-restricted multivariate regression. The component scores are the "x variables"; the component loadings are the regression coefficients. If the PCA was computed for a correlation matrix, you could think of the regression as being in standardized units. The fitted values are transformed from the standardized units back to the original units.

❑

❑ Technical note

You may have observed that the scoring coefficients were equal to the component loadings. This holds true for the principal components in unit normalization and for the orthogonal rotations thereof; it does not hold for oblique rotations.

❑

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():

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 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 loadings stores the following in r():

Macros
- r(cnorm) — component normalization: eigen, inveigen, or unit

Matrices
- r(A) — $p \times f$ matrix of normalized component loadings

estat residuals stores the following in r():

Matrices
r(fit) $p \times p$ matrix of fitted values
r(residual) $p \times p$ matrix of residuals

estat smc stores the following in r():

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

See the returned results of estat summarize in [R] estat summarize and of estat vce in [R] estat vce (available when vce(normal) is specified with pca or pcamat).

rotate after pca and pcamat add to the existing e():

Scalars
e(r_f) number of components 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_Ev) explained variance by rotated components

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

Methods and formulas

estat anti computes and displays the anti-image covariance matrix $\mathbf{C}$ and the anti-image correlation matrix $\mathbf{A}$

$$\mathbf{C} = \{\text{diag}(\mathbf{R})\}^{-1/2}\,\mathbf{R}\,\{\text{diag}(\mathbf{R})\}^{-1/2}$$

$$\mathbf{A} = \{\text{diag}(\mathbf{R})\}^{-1}\,\mathbf{R}\,\{\text{diag}(\mathbf{R})\}^{-1}$$

where $\mathbf{R}$ is the inverse of the correlation matrix.

estat kmo computes the "Kaiser–Meyer–Olkin measure of sampling adequacy" (KMO) and is defined as

$$\text{KMO} = \frac{\sum_{\mathcal{S}} r_{ij}^2}{\sum_{\mathcal{S}}(a_{ij}^2 + r_{ij}^2)}$$

where $\mathcal{S} = (i, j; i \neq j)$; r_{ij} is the correlation of variables i and j; and a_{ij} is the anti-image correlation. The variable-wise measure KMO_i is defined analogously as

$$\text{KMO}_i = \frac{\sum_{\mathcal{P}} r_{ij}^2}{\sum_{\mathcal{P}}(a_{ij}^2 + r_{ij}^2)}$$

where $\mathcal{P} = (j; i \neq j)$.

`estat loadings` displays the component loadings in different normalizations (see Jackson [2003, 16–18]; he labels them as **U**, **V**, and **W** vectors). Let $\mathbf{C} = \mathbf{L}\mathbf{\Lambda}\mathbf{L}'$ be the spectral or eigen decomposition of the analyzed correlation or covariance matrix $\mathbf{C}$, with $\mathbf{L}$ the orthonormal eigenvectors of $\mathbf{C}$, and $\mathbf{\Lambda}$ a diagonal matrix of eigenvalues. The principal components $\mathbf{A}$, that is, the eigenvectors $\mathbf{L}$, are displayed in one of the following normalizations:

cnorm(unit)	$\mathbf{A} = \mathbf{L}$	and so $\mathbf{A}'\mathbf{A} = \mathbf{I}$
normal(eigen)	$\mathbf{A} = \mathbf{L}\mathbf{\Lambda}^{1/2}$	and so $\mathbf{A}'\mathbf{A} = \mathbf{\Lambda}$
normal(inveigen)	$\mathbf{A} = \mathbf{L}\mathbf{\Lambda}^{-1/2}$	and so $\mathbf{A}'\mathbf{A} = \mathbf{\Lambda}^{-1}$

Normalization of the component loadings affects the normalization of the component scores.

The standard errors of the components are available only in unit normalization, that is, as normalized eigenvectors.

`estat residuals` computes the fitted values $\mathbf{F}$ for the analyzed correlation or covariance matrix $\mathbf{C}$ as $\mathbf{F} = \mathbf{L}\mathbf{\Lambda}\mathbf{L}'$ over the retained components, with $\mathbf{L}$ being the retained components in unit normalization and $\mathbf{\Lambda}$ being the associated eigenvalues. The residuals are simply $\mathbf{C} - \mathbf{F}$.

`estat smc` displays the squared multiple correlation coefficients SMC_i of each variable on the other variables in the analysis. These are conveniently computed from the inverse $\mathbf{R}$ of the correlation matrix $\mathbf{C}$,

$$\text{SMC}_i = 1 - \mathbf{R}^{-1}_{ii}$$

See [MV] **rotate** and [MV] **rotatemat** for details concerning the rotation methods and algorithms used.

The variance of the rotated loadings $\mathbf{L}_\text{r}$ is computed as $\mathbf{L}_{\text{r}'}\mathbf{C}\mathbf{L}_\text{r}$.

To understand `predict` after `pca` and `pcamat`, think of PCA as a fixed-effects factor analysis with homoskedastic residuals

$$\mathbf{Z} = \mathbf{A}\mathbf{L}' + \mathbf{E}$$

$\mathbf{L}$ contains the loadings, and $\mathbf{A}$ contains the scores. $\mathbf{Z}$ is the centered variables for a PCA of a covariance matrix and standardized variables for a PCA of a correlation matrix. $\mathbf{A}$ is estimated by OLS regression of $\mathbf{Z}$ on $\mathbf{L}$

$$\widehat{\mathbf{A}} = \mathbf{Z}\mathbf{B} \qquad \mathbf{B} = \mathbf{L}(\mathbf{L}'\mathbf{L})^{-}$$

The columns of $\mathbf{A}$ are called the scores. The matrix $\mathbf{B}$ contains the scoring coefficients. The PCA-fitted values for $\mathbf{Z}$ are defined as the fitted values from this regression, or in matrix terms,

$$\widehat{\mathbf{Z}} = \mathbf{Z}\mathbf{P}_\mathbf{L} = \mathbf{Z}\mathbf{L}(\mathbf{L}'\mathbf{L})^{-}\mathbf{L}'$$

with $\mathbf{P}_\mathbf{L}$ the orthogonal projection on (the rowspace of) $\mathbf{L}$.

This formulation allows orthogonal as well as oblique loadings $\mathbf{L}$ as well as loadings in different normalizations.

The above formulation is in transformed units. `predict` transforms the fitted values back to the original units. The component scores are left in transformed units, with one exception. After a PCA of covariances, means are substituted back in unless the option `centered` is specified. The residuals are returned in the original units. The residual sums of squares (over the variables) and the normalized versions are in transformed units.

References

Cattell, R. B. 1966. The scree test for the number of factors. *Multivariate Behavioral Research* 1: 245–276.

Jackson, J. E. 2003. *A User's Guide to Principal Components*. New York: Wiley.

Kaiser, H. F. 1974. An index of factor simplicity. *Psychometrika* 39: 31–36.

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

Also see

[MV] **pca** — Principal component analysis

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

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

[MV] **screeplot** — Scree plot

[U] 20 Estimation and postestimation commands