

Description

`scoreplot` produces scatterplots of the score variables after `factor`, `factormat`, `pca`, or `pcamat`, and scatterplots of the discriminant score variables after `discrim lda` or `candisc`.

`loadingplot` produces scatterplots of the loadings (factors or components) after `factor`, `factormat`, `pca`, or `pcamat`, and the standardized discriminant function loadings after `discrim lda` or `candisc`.

Quick start

Plot score variables

Plot score variables after a command such as `pca` or `factor`

```
scoreplot
```

Same as above, but add labels found in variable `mylabel`

```
scoreplot, mlabel(mylabel)
```

Same as above, but increase number of factors from 2 to 4

```
scoreplot, mlabel(mylabel) factors(4)
```

Same as above, but use unrotated results and combine plots instead of displaying matrix plot

```
scoreplot, mlabel(mylabel) factors(4) combined norotated
```

Plot loadings

Plot the loadings after a command such as `pca` or `factor`

```
loadingplot
```

Same as above, but increase the number of factors from 2 to 4

```
loadingplot, factors(4)
```

Same as above, but use unrotated results and combine plots instead of displaying matrix plot

```
loadingplot, factors(4) combined norotated
```

Menu

scoreplot

Statistics > Multivariate analysis > Factor and principal component analysis > Postestimation > Score variables plot

loadingplot

Statistics > Multivariate analysis > Factor and principal component analysis > Postestimation > Loading plot

Syntax

Plot score variables

```
scoreplot [ if ] [ in ] [ , scoreplot_options ]
```

Plot the loadings (factors, components, or discriminant functions)

```
loadingplot [ , loadingplot_options ]
```

<i>scoreplot_options</i>	Description
Main	
<code>factors(#)</code>	number of factors/scores to be plotted; default is <code>factors(2)</code>
<code>components(#)</code>	synonym for <code>factors()</code>
<code>norotated</code>	use unrotated factors or scores, even if rotated results exist
<code>matrix</code>	graph as a matrix plot, available only when <code>factors(2)</code> is specified; default is a scatterplot
<code>combined</code>	graph as a combined plot, available when <code>factors(# > 2)</code> ; default is a matrix plot
<code>half</code>	graph lower half only; allowed only with <code>matrix</code>
<code>graph_matrix_options</code>	affect the rendition of the matrix graph
<code>combine_options</code>	affect the rendition of the combined graph
<code>scoreopt(predict_opts)</code>	options for <code>predict</code> generating score variables
<code>marker_options</code>	change look of markers (color, size, etc.)
<code>marker_label_options</code>	change look or position of marker labels
Y axis, X axis, Titles, Overall	
<code>twoway_options</code>	any options other than <code>by()</code> documented in [G-3] <i>twoway_options</i>

<i>loadingplot_options</i>	Description
Main	
<code>factors(#)</code>	number of factors/scores to be plotted; default is <code>factors(2)</code>
<code>components(#)</code>	synonym for <code>factors()</code>
<code>norotated</code>	use unrotated factors or scores, even if rotated results exist
<code>matrix</code>	graph as a matrix plot, available only when <code>factors(2)</code> is specified; default is a scatterplot
<code>combined</code>	graph as a combined plot, available when <code>factors(# > 2)</code> ; default is a matrix plot
<code>half</code>	graph lower half only; allowed only with <code>matrix</code>
<code>graph_matrix_options</code>	affect the rendition of the matrix graph
<code>combine_options</code>	affect the rendition of the combined graph
<code>maxlength(#)</code>	abbreviate variable names to # characters; default is <code>maxlength(12)</code>
<code>marker_options</code>	change look of markers (color, size, etc.)
<code>marker_label_options</code>	change look or position of marker labels
Y axis, X axis, Titles, Overall	
<code>twoway_options</code>	any options other than <code>by()</code> documented in [G-3] <i>twoway_options</i>

Options

Main

`factors(#)` produces plots for all combinations of score variables up to `#`. `#` should not exceed the number of retained factors (components or discriminant functions) and defaults to 2. `components()` is a synonym. No plot is produced with `factors(1)`.

`norotated` uses unrotated results, even when rotated results are available. The default is to use rotated results if they are available. `norotated` is ignored if rotated results are not available.

`matrix` specifies that plots be produced using `graph matrix`; see [G-2] [graph matrix](#). This is the default when three or more factors are specified. This option may not be used with `combined`.

`combined` specifies that plots be produced using `graph combine`; see [G-2] [graph combine](#). This option may not be used with `matrix`.

`half` specifies that only the lower half of the matrix be graphed. `half` can be specified only with the `matrix` option.

`graph_matrix_options` affect the rendition of the matrix plot; see [G-2] [graph matrix](#).

`combine_options` affect the rendition of the combined plot; see [G-2] [graph combine](#). `combine_options` may not be specified unless `factors()` is greater than 2.

`scoreopt(predict_opts)`, an option used with `scoreplot`, specifies options for `predict` to generate the score variables. For example, after `factor`, `scoreopt(bartlett)` specifies that Bartlett scoring be applied.

`maxlength(#)`, an option used with `loadingplot`, causes the variable names (used as point markers) to be abbreviated to `#` characters. The `abbrev()` function performs the abbreviation, and if `#` is less than 5, it is treated as 5; see [FN] [String functions](#).

`marker_options` affect the rendition of markers drawn at the plotted points, including their shape, size, color, and outline; see [G-3] [marker_options](#).

`marker_label_options` specify if and how the markers are to be labeled; see [G-3] [marker_label_options](#).

Y axis, X axis, Titles, Overall

`twoway_options` are any of the options documented in [G-3] [twoway_options](#), excluding `by()`. These include options for titling the graph (see [G-3] [title_options](#)) and for saving the graph to disk (see [G-3] [saving_option](#)).

Remarks and examples

One of the main results from a principal component analysis, factor analysis, or a linear discriminant analysis is a set of eigenvectors that are called components, factors, or linear discriminant functions. These are stored in what is called a loading matrix. `pca`, `pcamat`, `factor`, and `factormat` store the loading matrix in `e(L)`. If there were p variables involved in the PCA or factor analysis, and f components or factors were retained, there will be p rows and f columns in the resulting loading matrix. `discrim lda` and `candisc` store the standardized canonical discriminant function coefficients or loadings in `e(L_std)`.

The columns of the loading matrix are in order of importance. For instance, with PCA, the first column of the loading matrix is the component that accounts for the most variance, the second column accounts for the next most variance, and so on.

In a loading plot, the values from one column of the loading matrix are plotted against the values from another column of the loading matrix. Of most interest is the plot of the first and second columns (the first and second components, factors, or discriminant functions), and this is what `loadingplot` produces by default. The rows of the loading matrix provide the points to be graphed. Variable names are automatically used as the marker labels for these points.

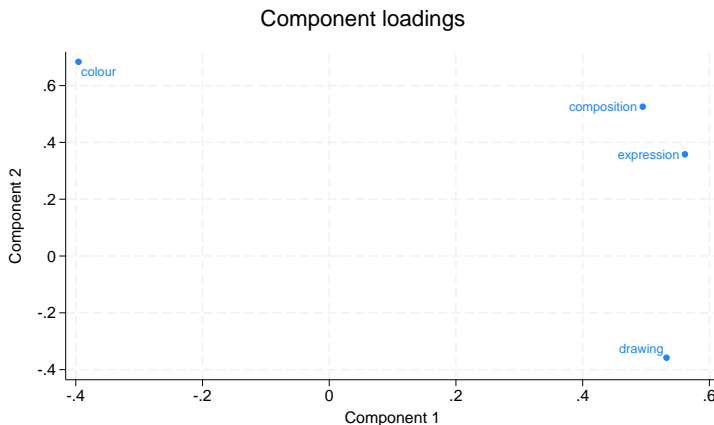
► Example 1: Loading plot

We use the Renaissance painters' data introduced in [example 2](#) of [\[MV\] biplot](#). There are four attribute variables recorded for 10 painters. We examine the first two principal component loadings for this dataset.

```
. use https://www.stata-press.com/data/r19/renpainters
(Scores by Roger de Piles for Renaissance painters)

. pca composition drawing colour expression
(output omitted)

. loadingplot
```



From the first component, we see that color (or colour, if you prefer) is separated from the other three attributes (variables): composition, expression, and drawing. From the second component, the difference in drawing stands out.

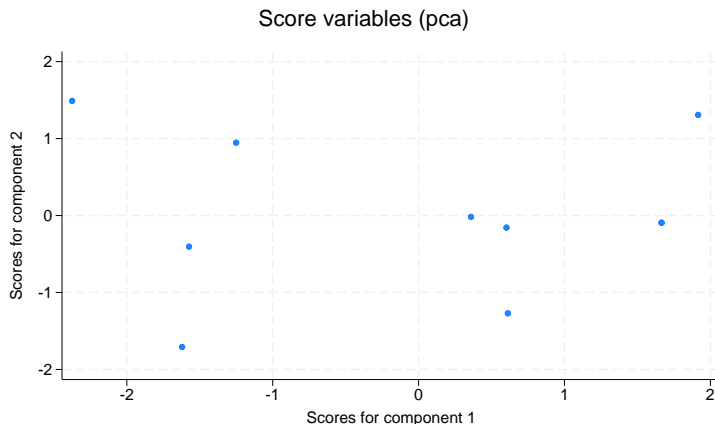
◀

Score plots approach the view of the loading matrix from the perspective of the observations. `predict` after `pca` and `factor` produces scores; see [\[MV\] pca postestimation](#) and [\[MV\] factor postestimation](#). `predict` after `discrim lda` and `candisc` can request discriminant function scores; see [\[MV\] discrim lda postestimation](#). A score for an observation from a particular column of the loading matrix is obtained as the linear combination of that observation's data by using the coefficients found in the loading. From the raw variables, the linear combinations described in the columns of the loading matrix are applied to generate new component or factor score variables. A score plot graphs one score variable against another. By default, `scoreplot` graphs the scores generated from the first and second columns of the loading matrix (the first two components or factors).

► Example 2: Score plot

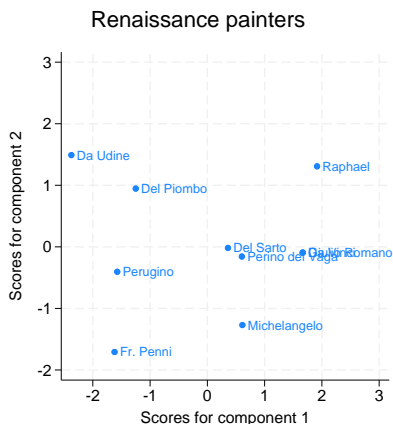
We continue with the PCA of the Renaissance painters.

```
. scoreplot
```



Unlike `loadingplot`, which can use the variable names as labels, `scoreplot` does not automatically know how to label the points after `factor` and `pca`. The graph above is not helpful. The *marker_label_option* `mlabel()` takes care of this.

```
. scoreplot, mlabel(painter) aspect(1) xlabel(-2(1)3) ylabel(-2(1)3)
> title(Renaissance painters)
```



We added a few other options to improve the graph. We extended the axes to include the value 3 so that there would be room for the marker labels, imposed an aspect ratio of 1, and added a title.

The score plot gives us a feeling for the similarities and differences between the painters. Da Udine is in an opposite corner from Michelangelo. The other two corners have Fr. Penni and Raphael.

You could refer to the loading plot and compare it with this score plot to see that the corner with Da Udine is most associated with the `colour` variable, whereas the corner with Michelangelo is best associated with drawing. Raphael is in the corner where the variables `composition` and `expression` are predominant.

If you like to make these kinds of associations between the variables and the observations, you will enjoy using the `biplot` command. It provides a joint view of the variables and observations; see [\[MV\] biplot](#).



After a rotation, the rotated factor or component loading matrix is stored in `e(r_L)`. By default, both `loadingplot` and `scoreplot` work with the rotated loadings if they are available. The `norotated` option allows you to obtain the graphs for the unrotated loadings and scores.

You can also request a matrix or combined graph for all combinations of the first several components or factors with the `components()` option (or the alias `factors()`).

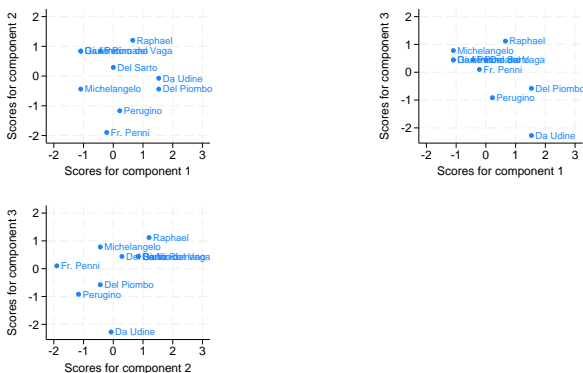
► Example 3: Score plots after a rotation

Even though the results from our initial look at the principal components of the Renaissance painters seem clear enough, we continue our demonstration by showing the score plots for the first three components after a rotation. See [\[MV\] rotate](#) for information on rotation in general, and see [\[MV\] pca postestimation](#) for specific guidance and warnings concerning rotation after a PCA.

```
. rotate
  (output omitted)

. scoreplot, mlabel(painter) components(3) combined
> aspect(.8) xlabel(-2(1)3) ylabel(-2(1)2)
```

Score variables (pca)



Rotation: orthogonal varimax

By default, the rotation information was included as a note in the graph. We specified `components(3)` to obtain all paired plots between the first three components. The `combined` option selected the combined view instead of the default matrix view of the graphs. The `aspect()`, `xlabel()`, and `ylabel()` options provide reasonable use of the graphing region while providing the same scale in the horizontal and vertical directions.



As the number of `factors()` or `components()` increases, the graphing area for each plot gets smaller. Although the default matrix view (option `matrix`) may be the most natural, the combined view (option `combined`) displays half as many graphs. However, the combined view uses more space for the labeling of axes than the matrix view. Regardless of the choice, with many requested factors or components, the graphs become too small to be of any use. In `loadingplot`, the `maxlength()` option will trim the

variable name marker labels that are automatically included. This may help reduce overlap when multiple small graphs are shown. You can go further and remove these marker labels by using the `mlabel("")` option.

Other examples of `loadingplot` and `scoreplot` are found in [\[MV\] pca postestimation](#), [\[MV\] factor postestimation](#), [\[MV\] discrim lda](#), and [\[MV\] discrim lda postestimation](#).

Also see

[\[MV\] candisc](#) — Canonical linear discriminant analysis

[\[MV\] discrim lda](#) — Linear discriminant analysis

[\[MV\] discrim lda postestimation](#) — Postestimation tools for discrim lda

[\[MV\] factor](#) — Factor analysis

[\[MV\] factor postestimation](#) — Postestimation tools for factor and factormat

[\[MV\] pca](#) — Principal component analysis

[\[MV\] pca postestimation](#) — Postestimation tools for pca and pcamat

[\[MV\] screeplot](#) — Scree plot of eigenvalues

