The *Stata Journal* publishes reviewed papers together with shorter notes or comments, regular columns, book reviews, and other material of interest to Stata users. Examples of the types of papers include 1) expository papers that link the use of Stata commands or programs to associated principles, such as those that will serve as tutorials for users first encountering a new field of statistics or a major new technique; 2) papers that go “beyond the Stata manual” in explaining key features or uses of Stata that are of interest to intermediate or advanced users of Stata; 3) papers that discuss new commands or Stata programs of interest either to a wide spectrum of users (e.g., in data management or graphics) or to some large segment of Stata users (e.g., in survey statistics, survival analysis, panel analysis, or limited dependent variable modeling); 4) papers analyzing the statistical properties of new or existing estimators and tests in Stata; 5) papers that could be of interest or usefulness to researchers, especially in fields that are of practical importance but are not often included in texts or other journals, such as the use of Stata in managing datasets, especially large datasets, with advice from hard-won experience; and 6) papers of interest to those who teach, including Stata with topics such as extended examples of techniques and interpretation of results, simulations of statistical concepts, and overviews of subject areas.

For more information on the *Stata Journal*, including information for authors, see the web page

http://www.stata-journal.com

The *Stata Journal* is indexed and abstracted in the following:

- CompuMath Citation Index®
- RePEc: Research Papers in Economics
- Science Citation Index Expanded (also known as SciSearch®)

**Copyright Statement:** The *Stata Journal* and the contents of the supporting files (programs, datasets, and help files) are copyright © by StataCorp LP. The contents of the supporting files (programs, datasets, and help files) may be copied or reproduced by any means whatsoever, in whole or in part, as long as any copy or reproduction includes attribution to both (1) the author and (2) the *Stata Journal*.

The articles appearing in the *Stata Journal* may be copied or reproduced as printed copies, in whole or in part, as long as any copy or reproduction includes attribution to both (1) the author and (2) the *Stata Journal*.

Written permission must be obtained from StataCorp if you wish to make electronic copies of the insertions. This precludes placing electronic copies of the *Stata Journal*, in whole or in part, on publicly accessible web sites, file servers, or other locations where the copy may be accessed by anyone other than the subscriber.

Users of any of the software, ideas, data, or other materials published in the *Stata Journal* or the supporting files understand that such use is made without warranty of any kind, by either the *Stata Journal*, the author, or StataCorp. In particular, there is no warranty of fitness of purpose or merchantability, nor for special, incidental, or consequential damages such as loss of profits. The purpose of the *Stata Journal* is to promote free communication among Stata users.

The *Stata Journal*, electronic version (ISSN 1536-8734) is a publication of Stata Press. Stata and Mata are registered trademarks of StataCorp LP.
Implementing Horn’s parallel analysis for principal component analysis and factor analysis

Alexis Dinno
Department of Biological Sciences
California State University–East Bay
Hayward, CA
adinno@post.harvard.edu

Abstract. I present paran, an implementation of Horn’s parallel analysis criteria for factor or component retention in common factor analysis or principal component analysis in Stata. The command permits classical parallel analysis and more recent extensions to it for the pca and factor commands. paran provides a needed extension to Stata’s built-in factor- and component-retention criteria.

Keywords: st0166, paran, parallel analysis, factor analysis, principal component analysis, factor retention, component retention, Horn’s criterion

1 Introduction

A method for factor or component retention is implemented in the Stata command paran, based on classical parallel analysis (Horn 1965) and recent Monte Carlo extensions to it (Glorfeld 1995). A critical aspect of principal component analysis (PCA) or factor analysis (FA) is the researcher’s decision of how many factors to retain. This decision can be motivated by a desire to reduce the analytic dimensionality of observed data, as when multiple scores are combined into one scale, or by a desire to unpack the structure underlying the covariance of observed data, as in exploratory FA (Velicer and Jackson 1990; Preacher and MacCallum 2003). While several criteria for retaining factors or components exist, a strong consensus has developed in the literature endorsing parallel analysis as among the most accurate methods (Montanelli and Humphreys 1976; Silverstein 1977; Zwick and Velicer 1986; Cota et al. 1993; Jackson 1993; Glorfeld 1995; Velicer, Eaton, and Fava 2000; Hayton, Allen, and Scarbello 2004; Lance, Butts, and Michels 2006). The choice of retention method is important, because different methods are more or less likely to overestimate or underestimate the number of factors or components.

Horn developed parallel analysis after considering the Kaiser rule, in which one retains eigenvalues greater than 1 for principal components, or greater than 0 for common factors (Kaiser 1960). Kaiser formulated this rule following a formal treatment by Guttman, demonstrating that in a population of $P$ variables of infinite size, eigenvalues greater than 1 form a theoretical lower bound on the number of components (or greater than 0 for factors) that can produce a correlation structure among the $P$ variables through linear combination (Guttman 1954). Put another way, a PCA of uncorrelated data would be expected to produce $P$ eigenvalues exactly equal to 1 in uncorrelated
Implementing Horn’s parallel analysis

data of infinite size. Horn reasoned that in a finite sample of size \( N \), one would expect to see eigenvalues greater than and less than 1 simply because of “sample bias”. Indeed, the poor performance of the Kaiser rule has resulted in its criticism in the methodological literature (Silverstein 1977; Zwick and Velicer 1986; Jackson 1993; Glorfeld 1995; Velicer, Eaton, and Fava 2000). Horn reasoned that this bias in the Kaiser rule could be corrected by generating a “sufficiently large” number, \( K \), of uncorrelated random data of the same number of \( N \) and \( P \) as the observed data; performing a parallel PCA or FA on each; and averaging the results. The bias estimate is thus this average eigenvalue minus 1 for each component (minus 0 for common factors). By subtracting this bias estimate from the eigenvalues from a PCA or FA on the observed data, one retains those adjusted eigenvalues greater than 1 (Horn 1965), or greater than 0 for common factors. More recently, it has been suggested that a more conservative approach would be to generate many random datasets (e.g., 5,000), and use the 95th or 99th percentile rather than the mean (Glorfeld 1995).

The \texttt{paran} command implements parallel analysis and Glorfeld’s extension to it. \texttt{paran} is a comprehensive command for parallel analysis, including the adaptation for FA, detailed reporting, graphing features including graphical representation of retained components, and Glorfeld’s (1995) Monte Carlo extension to parallel analysis. Stata’s user-written \texttt{fapara} command also implements a bare-bones parallel analysis computation after \texttt{pca} or \texttt{factor} has been run, with no options for textual or graphical output, and no implementation of the Monte Carlo variant. However, \texttt{fapara} makes its computations using Mata and may be faster.

2 Parallel analysis of data

2.1 Syntax

\texttt{paran} follows a syntax similar to \texttt{pca} and \texttt{factor}:

\begin{verbatim}
paran varlist [ if ] [ in ] [ weight ] [ , options ]
\end{verbatim}
### Options

<table>
<thead>
<tr>
<th>option</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Model</strong></td>
<td></td>
</tr>
<tr>
<td>iterations(#)</td>
<td>specify the number of iterations</td>
</tr>
<tr>
<td>centile(#)</td>
<td>specify using a centile value instead of the mean</td>
</tr>
<tr>
<td>factor(factor_type)</td>
<td>use FA instead of PCA; default is factor(pf)</td>
</tr>
<tr>
<td>citerate(#)</td>
<td>communality reestimation iterations</td>
</tr>
<tr>
<td><strong>Reporting</strong></td>
<td></td>
</tr>
<tr>
<td>quietly</td>
<td>suppress PCA or FA output</td>
</tr>
<tr>
<td>nostatus</td>
<td>suppress the status indicator</td>
</tr>
<tr>
<td>all</td>
<td>report all eigenvalues (default reports only those retained)</td>
</tr>
<tr>
<td><strong>Graphing</strong></td>
<td></td>
</tr>
<tr>
<td>graph</td>
<td>graph unadjusted, adjusted, and random eigenvalues</td>
</tr>
<tr>
<td>color</td>
<td>render graph in color (default is black and white)</td>
</tr>
<tr>
<td>lcolors(# # # # # # #)</td>
<td>specify colors using three RGB triples for observed, random, and adjusted eigenvalues (overrides the color option)</td>
</tr>
<tr>
<td>saving(filename)</td>
<td>save graph as a .gph file</td>
</tr>
<tr>
<td>replace</td>
<td>replace an existing file when saving()</td>
</tr>
<tr>
<td><strong>Miscellaneous</strong></td>
<td></td>
</tr>
<tr>
<td>protect(#)</td>
<td>perform # optimizations and report the best solution</td>
</tr>
<tr>
<td>seed(#)</td>
<td>seed the random-number generator with the supplied integer</td>
</tr>
<tr>
<td>copyleft</td>
<td>display the general public license for paran</td>
</tr>
</tbody>
</table>

- *fweights* and *aweights* are allowed; see [U] 11.1.6 weight.

### 2.2 Options

**Model options**

*iterations(#) sets the number of contrast datasets to evaluate.* The default value is $30 \times$ (the number of variables), and values less than 1 are ignored. For large datasets with large numbers of variables, many iterations can be time consuming. The greater the number of iterations, the more accurate the estimates of sample bias.
Implementing Horn’s parallel analysis

centile(#) specifies that the supplied centile value is to be used instead of the mean
(assumed median, because the distribution is symmetrical) in estimating bias. Values
above the mean/median, such as the 95th percentile, are more conservative estimates
of chance bias in the eigenvalues from a PCA of sample data. This option supercedes
the older pnf, which was equivalent to centile(95). Values of centile() must be
greater than 0 and less than 100. Noninteger values will be rounded to the nearest
integer value. Running paran without this option uses the mean value (very close
to centile(50)). See Glorfeld (1995).

factor(factor_type) selects one of the FA estimation types: pf, pcf, ipf, or ml (prin-
cipal factors, principal component factors, iterated principal factors, or maximum-
likelihood factors, respectively). If you specify anything but one of these four ab-
breviations, you will be warned and the program will halt. Note that conducting
parallel analysis using factor methods other than pf is unorthodox. Interpret such
results at your own risk. If factor() is not used, paran performs parallel analysis
using PCA by default.

citerate(#) sets how many iterations will be used to reestimate communalities for
the iterated principal factor type. citerate() can be used only with factor(ipf).

Reporting options

quietly suppresses output for the PCA or FA. This option is used only if a varlist is
specified in the paran command.

nostatus eliminates paran’s default behavior to indicate when every 10th percent of
the computation is complete.

all reports all components or factors, not just those with unadjusted eigenvalues greater
than 1 (or greater than 0 for FA). The default is to not report all components or
factors.

Graphing options

graph draws a graph of the observed eigenvalues, the random eigenvalues, and the
adjusted eigenvalues much like the graphs presented by Horn (1965).

color (use only with graph) renders the graph in color with unadjusted eigenvalues in
red, adjusted eigenvalues in black, random eigenvalues in blue, and all lines solid.
Without the color option, the graph is rendered in black and white, the line connect-
ing the unadjusted eigenvalues is dashed, the line connecting the random eigenvalues
is dotted, and the line connecting the adjusted eigenvalues is solid.
`lcolors(# # # # # # #)` (use only with `graph`) specifies the colors of each line on the graph with three RGB triples. The first triple indicates the R, G, and B components of the observed eigenvalues, the second triple sets the values for the mean or centile random eigenvalues, and the third triple sets the values for the adjusted eigenvalues. These settings override the default (red, blue, and black) colors of the `color` option.

`saving(filename)` (use only with `graph`) outputs the graph to the specified filename as a `.gph` file.

`replace` overwrites an existing filename when the `saving()` option is used with `graph`.

**Miscellaneous options**

`protect(#)` sets the number of optimizations for the starting values option for the maximum-likelihood factor type. `protect()` can be used only with `factor(ml)`.

`seed(#)` specifies an integer seed for the random-number generator (see `help set seed`) so that the results of `paran` for a specific dataset can be exactly reproduced. The default behavior of `paran` is to not specify a seed.

`copyleft` displays the copying permission statement for `paran`. `paran` is free software, licensed under GNU General Public License. The full license can be obtained by typing

```
. net describe paran, from(http://www.doyenne.com/stata)
```

and following the on-screen directions.

### 3 Saved results

The results of `paran` are returned in the matrix `r(HornEv)`, which is a $1 \times P$ matrix of the adjusted eigenvalues.

### 4 Example

A simulated dataset is included with this distribution. It contains 250 observations across 20 variables that have been defined by four random components plus an amount of noise unique to each measurement. The common variance in these data is constrained to 0.5 of the total variance. A classical parallel analysis of a PCA performed on these data can be obtained by typing

```
. use simdata
(Written by R. )
. paran X1-X20, all graph quietly seed(1)
```

Which, after a moment, produces the following output and figure 1.
Implementing Horn’s parallel analysis

Computing: 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%

Results of Horn’s Parallel Analysis for principal components
600 iterations, using the mean estimate

<table>
<thead>
<tr>
<th>Component</th>
<th>Adjusted Eigenvalue</th>
<th>Unadjusted Eigenvalue</th>
<th>Estimated Bias</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4.4965098</td>
<td>5.015828</td>
<td>.51507294</td>
</tr>
<tr>
<td>2</td>
<td>3.6971236</td>
<td>4.1506432</td>
<td>.4531958</td>
</tr>
<tr>
<td>3</td>
<td>2.4540371</td>
<td>2.8268943</td>
<td>.37285721</td>
</tr>
<tr>
<td>4</td>
<td>1.4124735</td>
<td>1.6603032</td>
<td>.24782968</td>
</tr>
<tr>
<td>5</td>
<td>.86086266</td>
<td>1.0739862</td>
<td>.21312356</td>
</tr>
<tr>
<td>6</td>
<td>.50264634</td>
<td>.69026757</td>
<td>.18762124</td>
</tr>
<tr>
<td>7</td>
<td>.41616116</td>
<td>.57773755</td>
<td>.14612135</td>
</tr>
<tr>
<td>8</td>
<td>.4171857</td>
<td>.48482857</td>
<td>.06764317</td>
</tr>
<tr>
<td>9</td>
<td>.41289346</td>
<td>.46305685</td>
<td>.05016339</td>
</tr>
<tr>
<td>10</td>
<td>.38351215</td>
<td>.41568018</td>
<td>.03216803</td>
</tr>
<tr>
<td>11</td>
<td>.42688028</td>
<td>.38397364</td>
<td>-.04290664</td>
</tr>
<tr>
<td>12</td>
<td>.39338348</td>
<td>.34201688</td>
<td>-.05018181</td>
</tr>
<tr>
<td>13</td>
<td>.45403937</td>
<td>.3178023</td>
<td>-.13625914</td>
</tr>
<tr>
<td>14</td>
<td>.45767627</td>
<td>.29589049</td>
<td>-.16178578</td>
</tr>
<tr>
<td>15</td>
<td>.44264261</td>
<td>.25826351</td>
<td>-.1843791</td>
</tr>
<tr>
<td>16</td>
<td>.49499729</td>
<td>.24047616</td>
<td>-.25452113</td>
</tr>
<tr>
<td>17</td>
<td>.51687283</td>
<td>.23139579</td>
<td>-.28547704</td>
</tr>
<tr>
<td>18</td>
<td>.54348581</td>
<td>.20336817</td>
<td>-.34011763</td>
</tr>
<tr>
<td>19</td>
<td>.55823785</td>
<td>.18720304</td>
<td>-.3710348</td>
</tr>
<tr>
<td>20</td>
<td>.64292367</td>
<td>.18346663</td>
<td>-.45945704</td>
</tr>
</tbody>
</table>

Criterion: retain adjusted components > 1

Figure 1. A plot showing the results of the parallel analysis of a PCA on simulated data with four true components underlying 20 variables. Nonretained components are marked with a hollow circle on the adjusted eigenvalues curve.
By specifying the seed for Stata’s random-number generator, users can reproduce these results exactly. The default uses 30 times the number of variables iterations, or 600 here. We see in the results that four adjusted eigenvalues are greater than 1. A graph is produced, as shown in figure 1. The dashed line plots the unadjusted eigenvalues in decreasing order, in a “scree plot” as per Cattell (1966). The dotted line is the scree plot for the mean eigenvalues of random $N$ by $P$ data. The adjusted eigenvalues are plotted with a solid line. Horn’s criterion corresponds to the point where the adjusted eigenvalues cross the horizontal line at $y = 1$, which is equivalent to the point where the unadjusted eigenvalues cross the curve of mean random eigenvalues. This example demonstrates the sometimes different results given by parallel analysis versus Kaiser’s “eigenvalue greater than 1” rule for the number of components or factors to retain. The former obtained the correct number of components, while the latter overestimated the number of components.

5 Technical notes

Horn (1965) suggested that the simulated datasets be normal with means of 0 and unit variance. Thompson and Daniel (1996) asserted that data for the simulation be of the same “rank” as the observed data. More recently, Hayton, Allen, and Scarpello (2004) urged a parameterization of the random data to approximate the distribution of the observed data with respect to the middle (midpoint) and the observed minimum and maximum. However, PCA and FA standardize each variable to describe the total and common variance, respectively, so any linear transformation of all variables produce the same eigenvalues. This is born by the notable lack of difference between analyses conducted using a variety of simulated distributional assumptions (Dinno Forthcoming). The central limit theorem would seem to make the selection of a distributional form for the random data moot with any sizeable number of iterations.

6 References


Implementing Horn’s parallel analysis


About the author

Alexis Dinno is a social epidemiologist and social ecologist with a strong interest in applied quantitative methods. She presently lectures part time at California State University–East Bay, and she conducts research in tobacco control policy and the links between urban neighborhood conditions and residents’ depressive experiences. She has an abiding interest in the links between applied research methods and theory.