Identifying or Verifying the Number of Factors to Extract using Very Simple Structure.

Link to the last RSS article here: Statistical Resources (update; version 3), -- Ed.

By Dr. Jon Starkweather, Research and Statistical Support Consultant Team

Factor analysis is perhaps one of the most frequently used analyses. It is versatile and flexible; meaning, it can be applied to a variety of data situations and types, and it can be applied in a variety of ways. However, conducting factor analysis generally requires the data analyst to make several decisions. Analysts often run several factor analyses, even when attempting to confirm an established factor structure; in order to assess the fit of the data to several factor models (e.g. one factor model, two factor model, three factor model, etc.). Over the 100 years since Spearman (1904) developed factor analysis there have been many, many criteria proposed for determining the number of factors to extract (e.g. eigenvalues greater than one, Horn's [1965] parallel analysis, Cattell's [1966] scree plot or test, Velicer's [1976] Minimum Average Partial [MAP] criterion, etc.). Each of these proposed criteria have strengths and weaknesses; and they occasionally conflict with one another, which makes using one criterion over another a risky proposition. This month's article demonstrates a very handy method for comparing multiple criteria in the pursuit of choosing to extract the appropriate number of factors during factor analysis.

In popular culture it is not uncommon to hear someone say, "There's an app for that." The phrase generally refers to the idea that an application exists (for a smart phone) which does the task being discussed. Likewise, here at RSS we very frequently find "There's a pack for that." This phrase refers to the virtual certainty of finding an R package which has a function devoted to some analysis or technique we are discussing. The primary package we will be using here is one package which contains a great many useful functions and as a result is very often the package we end up using for a variety of analyses. The primary package we will be using here is the 'psych' package (Revelle, 2014). The 'psych' package has grown substantially over the last few years and includes many very useful functions – if you have not taken a look at it recently, you might want to check it out.

Our examples below will actually require two packages, the 'psych' package and the 'GPArotation' package (Bernaards & Jennrich, 2014). The 'GPArotation' package should be familiar to anyone with experience doing factor analysis – it provides functions for several rotation strategies. The primary function we demonstrate below is the 'vss' function from the 'psych' package. The Very Simple Structure (VSS; Revell & Rocklin, 1979) function provides a nice output of criteria for varying levels of factor model complexity (i.e. number of factors to extract). The Very Simple Structure (VSS) terminology is used to refer to the idea that all loadings which are less than the maximum loading (of an item to a factor) are suppressed to zero – thus forcing a particular factor model to be much more interpretable or more clearly distinguished. Then, fit of several models of increasing rank complexity (i.e. more and more factors specified) can be assessed using the residual matrix of each model (i.e. original matrix minus the reproduced matrix of the models). We will also be using both the 'fa' function (from the 'psych' package) and the 'factanal' function (from the 'stats' package – included with all installations of R) to fit factor analysis models to the data structures.

Examples
The first two examples used here can easily be duplicated using the scripts provided below (i.e. the data file is available at the URL in the script / screen capture image). The third example is the example contained in the help file of the ‘vss’ function and can be accessed using the script below. First, load the two packages we will be using.

```r
library(psych)
library(ORaCulation)
```

Next, we will import the comma delimited text (.txt) file from the RSS server using the URL and file name (vss_df.txt) contained in the script / image below. We also run a simple ‘summary’ on the data frame to make sure it was imported correctly.

The simulated data includes a sample identification number for each participant (s.id), a grouping variable (group 1 or group 2), age of each participant (age in years), sex of each participant (female or male), class standing of each participant (freshman, sophomore, junior, or senior), and 30 item scores. Next, we will identify which participants belong to group 1 and which belong to group 2; as well as the number of participants in each group.

So, we have 418 participants in group 1 and 982 participants in group 2. Generally when analysts intend to do factor analysis they have an idea of how many factors they believe the appropriate factor model contains; and often they have an idea of whether an orthogonal or oblique rotation strategy is warranted. For this first example (i.e. group 1) looking at the 30 item scores (i.e. columns 6 through 35), we believe there are two factors and therefore; we specify 3 factors (n = 3) in the ‘vss’ function. We also believe the factors are likely to be meaningfully related and...
consequently, we specify an oblimin rotation strategy. Next, we apply the ‘vss’ function to group 1. Also note, we specified Maximum Likelihood Estimation as the Factor Method (fm = "mle") because this is the method used by default with the ‘factanal’ (i.e. factor analysis) function of the ‘stats’ package. We specified the number of observations (i.e. number of rows, cases, or participants) using the length of the group 1 vector (g1). Recall from above, the group 1 vector contains the row numbers of all the participants from group 1.

The first few rows of output (i.e. "Very Simple Structure" table) show the function called and the maximum complexity values. This is a good example because the VSS complexity rows are conflicting; VSS complexity 1 shows a 2-factor model is best while VSS complexity 2 indicates a 3-factor model is best. The VSS complexity 2 is a bit misleading because both the 2-factor model and 3-factor model display a VSS complexity 2 of 0.80; as can be seen in the first column of output under the "Statistics by number of factors" table. So, in fact both complexity 1 and complexity 2 are in agreement. Furthermore, the Velicer MAP minimum is reached with the 2-factor model; which can also be seen in the third column of the "Statistics by number of factors" table. The Bayesian Information Criterion (BIC) minimum is reached with the 2-factor model; as well as the Sample Size adjusted BIC (SABIC) – shown in columns 10 and 11 respectively of the "Statistics by number of factors" table. The ‘vss’ function also produces a plot (by default) which shows the number of factors on the x-axis and the VSS (complexity) Fit along the y-axis with lines and numbers in the Cartesian plane representing the (3) different factor models (see below).

To interpret the graph, focus on the model (1, 2, or 3 factor models) which has the highest line (and numerals) in relation to the y-axis; but also note any transitions of the model lines. In this example, the transitions are all very nearly flat but a later example will better demonstrate the utility of this type of plot.

Next, we can verify the fit of our 2-factor model using either the ‘fa’ function (from the ‘psych’ package) and / or the ‘factanal’ function (of the ‘stats’ package).
We will now assess the group 2 (g2) data. This group is believed to be best served with a 3-factor model; so we specify 4 factors \(n = 4\) in the ‘vss’ function call; again with the factor method set to Maximum Likelihood Estimation (\(fm = \text{"mle"}\)) and an oblique rotation strategy (\(rotate = \text{"oblimin"}\)).
In this example all of the indices in the top table ("Very Simple Structure") are in agreement; although both VSS complexity metrics display the same maximum for a 3-factor model and a 4-factor model. Looking at the first two columns of the "Statistics by number of factors" table shows the identical complexity maximums (0.84) for both the 3-factor model (row 3) and the 4-factor model (row 4) with both complexities 1 and 2 (columns 1 and 2). But, given the other indices agreement in support of the 3-factor model, that would be the model most appropriate. The plot (below) reinforces the interpretation of the tabular output above.

The plot (above) shows that the 3-factor model is meaningfully better than the 1-factor or 2-factor models and the 4-factor model does not show any improvement over the 3-factor model – which is evident because the number 4 in the plot is not [further] above the line associated with the 3-factor model (i.e. no gain or transition upward; as is the case from 1-factor to 2-factors and to 3-factors). Therefore, we fit the 3-factor model to our data using the ‘fa’ function (of the ‘psych’ package) and / or the ‘factanal’ function of the ‘stats’ package.
The next example is straight from the help file of the 'vss' function and is discussed here because it demonstrates a situation when the tables of output from the 'vss' function are not in agreement. When this situation occurs, one must rely upon the plot produced by the 'vss' function rather than the textual output. First, open the help file (here the plain text version is shown).

*Note: the last few lines of output from the 'fa' function are cut off (i.e. not shown).

*Note: last few lines of output from the 'factanal' function are cut off (i.e. not shown).
Next, scroll to the bottom of the help file and copy / paste the relevant lines of script into the R console.

As mentioned previously, the tables of statistics do not provide a clear answer to the question of which factor model is best (i.e. how many factors should be extracted). However, if we review the associated plot, we can clearly see the 4-factor model is the best (i.e. highest; even when embedded within models with more than 4 factors, with good separation from previous models).
Conclusions

The intent of this article was to raise awareness of the dangers of using only one criteria or method for deciding upon the number of factors to extract when conducting factor analysis. This article also demonstrated the ease with which an analyst can compute and evaluate several such criteria to reach a more informed decision. More extensive examples of the data analysis solutions are available at the RSS Do-it-yourself Introduction to R course page. Lastly, a copy of the script file used for the above examples is available here.

Until next time; remember what George Carlin said: "Just 'cause you got the monkey off your back doesn't mean the circus left town."

References / Resources


Originally published December 2014 -- Please note that information published in *Benchmarks Online* is likely to degrade over time, especially links to various Websites. To make sure you have the most current information on a specific topic, it may be best to search the UNT Website - http://www.unt.edu. You can also consult the UNT Helpdesk - http://www.unt.edu/helpdesk/. Questions and comments should be directed to benchmarks@unt.edu.

Contact Us:

**University Information Technology**
1155 Union Circle #310709
Denton, TX 76203 USA
Voice: 940-565-4068
Fax: 940-565-4060

Visit Us:
Sage Hall, Room 338
http://it.unt.edu/benchmarks/

Email us:
Have questions on content or technical issues? Please contact us. unt.uit@unt.edu

UNT System:
- UNT Home
- UNT System
- UNT Dallas
- UNT Health Science Center

Site last updated on April 22, 2016

Disclaimer | AA/EOE/ADA | Privacy Statement | Web Accessibility Policy | State of Texas Online | Emergency Preparedness