Factor Analysis

-Applied Multivariate Analysis-

Lecturer: Darren Homrighausen, PhD

From PCA to factor analysis

Remember: PCA tries to estimate a transformation of the data such that:

1. The maximum amount of variance possible is explained by each component
2. The components are uncorrelated (equivalently, they are perpendicular)

We are going to motivate an algorithm for finding a different transformation of the data via two stories.

First story: Measurement error

Suppose the the numbers we write down into our $\mathbb{X}$ matrix aren't accurate.

[That is, there is measurement error in the explanatory variables]

PCA seeks to reproduce all the variance in $\mathbb{X}$ in the fewest number of components

This is odd, as we want to keep the signal variance, but would like to get rid of the noise variance.

First story: Measurement error

This situation in notation would be:

$$X_j = \underbrace{\mu_j}_{signal} + \underbrace{\epsilon_j}_{noise}$$

Example: We are recording diameters of trees at a certain height to answer questions about the success of various tree species. Sometimes, the measuring tape will not be level. Maybe the measuring tape only has inches and the experimenter just guesses at the remaining fraction. The true diameter would be μ_j and these other experiment issues would be ϵ_j.

Second story: Maintaining correlation

(Without getting into too many details)

Factor analysis (FA) finds a small number of factors that maintain the most amount of correlations from the original data, instead of the variance (which would be PCA)

To do this, we will seek to find latent variables or latent factors.

What is a latent variable?

Definition: A latent variable (or factor) is an unmeasured, underlying feature that actually is driving your observation

Example of a latent variable

Suppose I record all the driving routes you take from your house to another location and then back to your house

(Imagine the destination location isn't recorded)

There would be many such routes, and many of them would be similar

However, there would probably be several different routes to the same general location

(You won't go to exactly same location as you'll park in different parking spaces, etc.)

Suppose one group of routes all ended near a grocery store. Then I could say there is a latent variable for some of your movements that was "going to get food/supplies"

Roots of factor analysis: Causal discovery

Charles Spearman (1904) hypothesized a hidden structure for intelligence

He observed that schoolchildren's grades in different subjects were all correlated with each other

He thought he could explain this by reasoning that grades in different subjects are correlated because they are all functions of some common factor: 'general intelligence,' which he notated as g or G

[interestingly enough, intelligence is *still* referred to as g or G. Let this be a lesson: choose your notation carefully. It might be used for a very long time]

Roots of factor analysis: Causal discovery

Spearman's model was

$$X_{ij} = G_i W_j + \epsilon_{ij}$$

where

- i indexes children in the study
- j indexes school subjects
- ϵ_{ij} is uncorrelated noise
- G_i is the i^{th} child's intelligence value. Think of G as a function evaluated at that person, giving her some amount of intelligence
- W_j is the amount subject j is influenced by G.

The idea: Given a value of intelligence G_i, a person's grades are uncorrelated and merely a (random) function of how much intelligence affects achievement in a subject

Roots of factor analysis: Causal discovery

Spearman concluded based on some experiments that a single factor G existed

Later research using large batteries of tests demonstrated that this does not seem to be the case in general.

Spearman and other researchers decided this meant that one factor wasn't enough. Thus factor analysis was born.

Factor analysis (FA)

With multiple factors, Spearman's model becomes

$$(X_{ij} = G_i W_j + \epsilon_{ij}) \quad \Rightarrow \quad \left(X_{ij} = \sum_{k=1}^{K} F_{ik} W_{kj} + \epsilon_{ij} \right)$$

where

- the factors F_k are mean zero, and unit variance
- ϵ_{ij} are uncorrelated with each other and the factors F_k.
- The $X_{.j}$ are mean zero

  (That is; the covariates have already had their mean subtracted off: i.e: $\mathbb{X} - \overline{\mathbb{X}}$)

Factor analysis (FA)

$$X_{ij} = \sum_{k=1}^{K} F_{ik} W_{kj} + \epsilon_{ij} \qquad \Leftrightarrow \qquad \mathbb{X} = FW + \epsilon$$

Some terminology:

- The F_{ik} are the factor scores of the i^{th} observation on the k^{th} factor
- The W_{kj} are the factor loadings of the j^{th} explanatory variable on the k^{th} factor.

Compare to PCA ($\mathbb{X} = UDV^\top = SV^\top$):

$$X_{ij} = \sum_{k=1}^{p} (UD)_{ik} V_{kj} = \sum_{k=1}^{p} S_{ik} V_{jk}$$

There's a problem...

Reminder: we call a matrix O orthogonal if its inverse is its transpose:

$$O^\top O = I = OO^\top$$

[think about the SVD, where $\mathbb{X} = UDV^\top$ and $V^\top V = I$]

So, with our model

$$\mathbb{X} = FW + \epsilon = FOO^\top w + \epsilon = F'w' + \epsilon$$

where $F' = FO$ and $W' = O^\top W$.

Conclusion: *we changed the factors scores and loadings, but the data didn't change at all*

(In statistics, we can such a situation unidentifiable)

Note: This doesn't happen with PCA

$$\mathbb{X} = UDV^\top = UDOO^\top V^\top = U \underbrace{D'}_{\text{not diagonal}} V'^\top$$

Factor analysis model

Note that, for any i and each j

$$
\begin{aligned}
\sigma_j^2 = \mathrm{Var}(X_{ij}) &= \mathrm{Var}\left(\sum_{k=1}^{K} F_{ik} W_{kj} + \epsilon_{ij}\right) \\
&= \sum_{k=1}^{K} \mathrm{Var}\left(F_{ik} W_{kj}\right) + \mathrm{Var}\left(\epsilon_{ij}\right) \\
&= \sum_{k=1}^{K} W_{kj}^2 + \mathrm{Var}\left(\epsilon_{ij}\right) \\
&= \underbrace{\sum_{k=1}^{K} W_{kj}^2}_{\text{Factors' variance}} + \underbrace{\psi_j}_{\text{Error variance}}
\end{aligned}
$$

Factor analysis model

$$\sigma_j^2 = \mathrm{Var}(X_{ij}) = \underbrace{\sum_{k=1}^{K} W_{kj}^2}_{\text{Factors' variance}} + \underbrace{\psi_j}_{\text{Error variance}}$$

This decomposes the variance of each explanatory variable:

- $\sum_{k=1}^{K} W_{kj}^2$ is known as the communality
- ψ_j is known as the specific variance

Factor analysis model

Note that additionally,

$$\sigma_{jl}^2 = \mathrm{Cov}(X_{ij}, X_{il}) = \sum_{k=1}^{K} W_{kj} W_{kl}$$

This implies

$$\begin{aligned}\Sigma &= \begin{bmatrix} \sigma_1^2 & \cdots & \sigma_{1p}^2 \\ & \vdots & \\ \sigma_{p1}^2 & \cdots & \sigma_p^2 \end{bmatrix} \\ &= \begin{bmatrix} \sum_{k=1}^{K} W_{k1}^2 + \psi_1 & \cdots & \sum_{k=1}^{K} W_{k1} W_{kp} \\ & \vdots & \\ \sum_{k=1}^{K} W_{kp} W_{k1} & \cdots & \sum_{k=1}^{K} W_{kp}^2 + \psi_p \end{bmatrix} \\ &= W^\top W + \Psi \end{aligned}$$

(Ψ as the diagonal matrix with entries ψ_j)

Factor analysis model: Our goal

Now that we have the induced equation of the covariances:

$$\Sigma = W^\top W + \Psi$$

we want to estimate W such that for a given K

- $W^\top W$ is as large as possible
- Ψ is a small as possible

(One note: we never observe Σ. We need to estimate it with $\hat{\Sigma} = \frac{1}{n}\mathbb{X}^\top\mathbb{X}$)

Find Ψ & W: Maximum likelihood

In order to get at the W, we need a good estimate of Ψ

One way is through maximum likelihood (ML).

This has a cost: making assumptions about the distribution of F:

$$F_{ik} \overset{i.i.d.}{\sim} N(0,1)$$

which implies

$$X_{i\cdot} \overset{i.i.d.}{\sim} N(0, \Psi + W^\top W)$$

Hence, the closer $\Psi + W^\top W$ is to $\hat{\Sigma}$, the higher the likelihood.
(The details of the maximization are tedious and we won't discuss them here)

(Note that implicit in ML is a likelihood ratio test for K. We will discuss this later when covering methods for choosing K)

Find Ψ & W: Principal factor analysis

Principal factor analysis (PA) has strong ties to PCA. The difference is:

1. Guess a $\hat{\Psi}$
2. Form the reduced covariance matrix $\hat{\Sigma}^* = \hat{\Sigma} - \hat{\Psi}$. Hence, the diagonals are the communalities we wish to estimate.

   (Remember, we are trying to solve $\hat{\Sigma} = W^\top W + \Psi$)
3. Form $\hat{\Sigma}^* = V' D'^2 V'^\top$
4. Define W_j to be the j^{th} row of the matrix V with only its first K columns

   (These would be the PC loading vectors if $\hat{\Sigma}^*$ was the covariance matrix)
5. Re-estimate $\hat{\Psi} = \hat{\Sigma} - \hat{W}^\top \hat{W}$
6. Keep returning to step 1. until $\hat{\Psi}$ and $\hat{W}$ don't change much between cycles

What about estimating F?

In PCA, arguably the most important quantities were the principal component scores given by UD.

These are the coordinates of the data expressed in the PC basis

Of somewhat lesser importance are the principal component loadings, given by the rows of the matrix V.

In factor analysis, the factor loadings are everything.

It is much more important in most applications to find how the covariates load on the latent factors than to find how the observations are expressed in those factors (the factor scores)

We can still estimate F (by least squares), but it isn't usually considered a useful quantity. This is application specific, however

Uses of factor analysis

Factor analysis can be used for two major things:

- Exploratory factor analysis: (EFA) To identify complex interrelationships among variables that are part of unified concepts. The researcher makes no a priori assumptions about relationships among factors
- Confirmatory factor analysis (CFA): To test the hypothesis that the items are associated with specific factors

Usually, you split your data in two parts and run EFA on one part and test your conclusion using CFA on the other part

Factor analysis in R

Like usual, someone has done the heavy lifting for you and made a nice R package called psych

```
library(psych)
Xcorr  = cor(X)
out    = fa(r=Xcorr,nfactors=nfactors,
              rotate='none',fm='pa')
```

Let's discuss this a bit:

- r=Xcorr: Factor analysis operates on the correlation matrix. Alternatively, we could pass the original data $\mathbb{X}$
- nfactors: We need to specify the number of factors we are looking for (we'll return to this later)
- rotate = 'none': This controls which O we used

  (That is, the orthogonal matrix O inside of $FOO^\top W$)
- fm='pa': Method to get the W. Can use 'ml' or 'minres'

METHOD PARAMETER IN FA

The solution depends strongly on the estimation procedure

I'm not aware of a detailed comparison of these two methods

My recommendation:

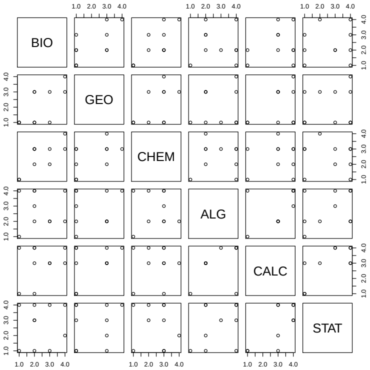
- PA is preferred as it relies on weaker assumptions. However, it can sometimes produce $\hat{\psi}_j$ that are negative, which corresponds to negative variance. If this happens, use ML
- The ML solution comes with more features, but it requires much stronger assumptions that are hard to check. Additionally, the maximization doesn't always converge.
- If both PA and ML fail, use minimum residuals (known as minres in R

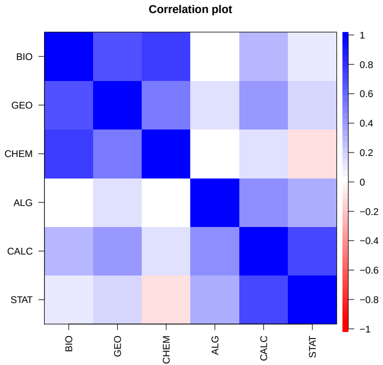
Let’s look at an example

```
library(psych)
X = read.table(’../data/gradesFA.txt’,header=T)
Xcorr  = cor(X)

> round(Xcorr,2)
       BIO  GEO  CHEM   ALG CALC  STAT
BIO   1.00 0.66  0.74 -0.02 0.27  0.09
GEO   0.66 1.00  0.53  0.10 0.40  0.16
CHEM  0.74 0.53  1.00  0.00 0.10 -0.13
ALG  -0.02 0.10  0.00  1.00 0.42  0.32
CALC  0.27 0.40  0.10  0.42 1.00  0.72
STAT  0.09 0.16 -0.13  0.32 0.72  1.00
```

Alternate visualizations





```
##Left plot
pairs(X)
##Right plot
cor.plot(Xcorr)  #in psych package
```

Let's look at an example (Output)

```
out.fa = fa(X,nfactors=2,rotate='none',fm='pa')
> out.fa
Factor Analysis using method =  pa
[omitted]
Standardized loadings
      PA1   PA2   h2    u2
BIO  0.81 -0.46 0.86 0.142
GEO  0.70 -0.19 0.53 0.470
CHEM 0.61 -0.54 0.67 0.330
ALG  0.24  0.36 0.19 0.815
CALC 0.73  0.66 0.98 0.023
STAT 0.42  0.63 0.57 0.429
```

Here $K = 2$, and

- The 'Standardized loadings' are the entries in W
- h2 are the communality $(\sum_{k=1}^{K} W_{kj}^2)$
- u2 are specific variances $(\hat{\psi}_j)$

Let's look at an example (Output continued)

```
                       PA1  PA2
SS loadings           2.28 1.51
Proportion Var        0.38 0.25
Cumulative Var        0.38 0.63
[omitted]
```

These are

- SS loadings: The sum of the squared loadings on each factor
- Proportion Var: The amount of total variation explained by each factor

Return to the loadings table

```
Standardized loadings
      PA1   PA2   h2    u2
BIO  0.81 -0.46 0.86 0.142
GEO  0.70 -0.19 0.53 0.470
CHEM 0.61 -0.54 0.67 0.330
ALG  0.24  0.36 0.19 0.815
CALC 0.73  0.66 0.98 0.023
STAT 0.42  0.63 0.57 0.429
```

We would like this solution to be ‘clean.’ This means

- each explanatory variable is highly loaded on only one factor
- all loadings are either large or near zero

  (Zero mean something like $<$ .1 in absolute value)

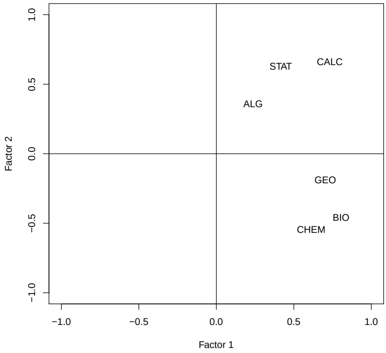
This solution has neither

(Actually, `rotation=‘none’` very commonly produces ‘unclean’ solutions)

Graphical representation

For a two factors solution, this can be performed graphically

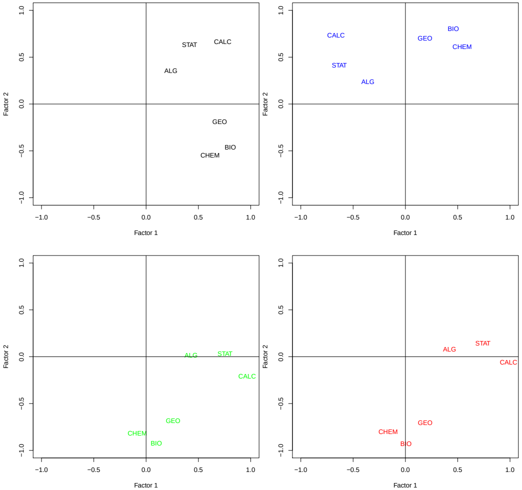
```
Standardized loadings
      PA1   PA2   h2    u2
BIO  0.81 -0.46 0.86 0.142
GEO  0.70 -0.19 0.53 0.470
CHEM 0.61 -0.54 0.67 0.330
ALG  0.24  0.36 0.19 0.815
CALC 0.73  0.66 0.98 0.023
STAT 0.42  0.63 0.57 0.429
```



Example rotations

Remember: the O matrix was arbitrary!

We can make any $W' = O^\top W$ we want



Black: 'none'

Blue: A rotation I made up that is bad

Green: A rotation I made up that is good

Red: A non-orthogonal rotation I made up that is the best

Automated rotations?

Any solution we estimate is just one of an infinite number of solutions

(Given by all possible rotations $W' = O^\top W$)

Manually trying many rotations becomes impossible if there are more than 2 factors or if p is not small

We need ways to choose good O automatically

There are many proposed methods for choosing the rotation

- Varimax rotation
- Quartimax rotation
- Equimax rotation
- Direct oblimin rotation
- Promax rotation

Automated rotations?

Any solution we estimate is just one of an infinite number of solutions

(Given by all possible rotations $W' = O^\top W$)

Manually trying many rotations becomes impossible if there are more than 2 factors or if p is not small

We need ways to choose good O automatically

There are many proposed methods for choosing the rotation

- Varimax rotation
- Quartimax rotation
- Equimax rotation
- Direct oblimin rotation
- Promax rotation

Varimax and Oblimin rotation

Varimax: Finds an O that maximizes the variance of the diagonals of $W^\top W$ while keeping the latent factors uncorrelated

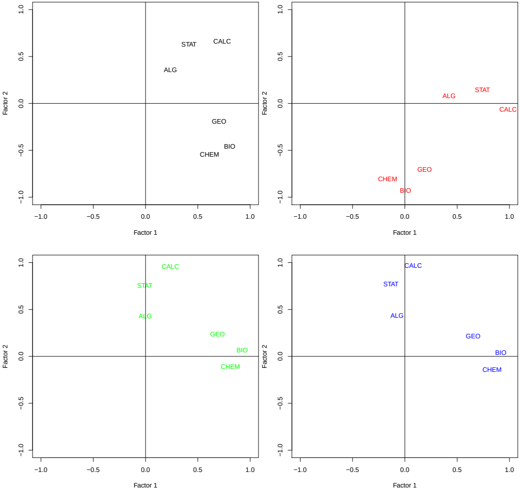
Advantage: This has the advantage of making the W either large or small, which will help in interpretability

Keeping the factors uncorrelated is often unrealistic. This leads to...

Direct oblimin rotation: If we consider correlated factors, this leads to non-orthogonal (known as oblique) rotations. Introducing correlations complicates the estimation process. However, it often leads to more interpretable factor solutions.

Conclusion: Direct oblimin rotation is the standard method, but it leads to (some) complications relative to

Automated rotations



Black: ‘none’

Red: A non-orthogonal rotation I made up

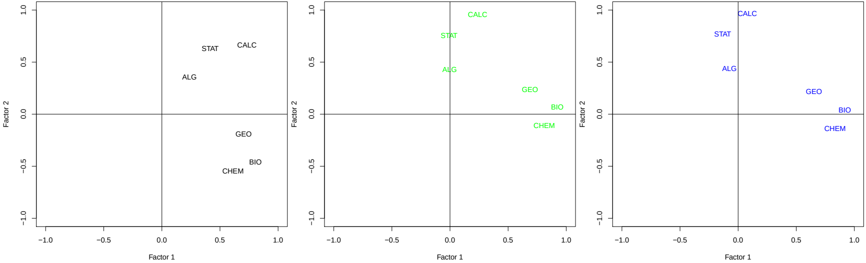
Green: Varimax rotation

Blue: Oblimin oblique rotation

(Need package GPArotation)

Comparison of factor solution

```
           None              Varimax          Oblimin
BIO   0.80 -0.45     |     0.92          |     0.91
GEO   0.70 -0.18     |     0.68  0.23     |     0.65  0.21
CHEM  0.61 -0.54     |     0.81 -0.10     |     0.83 -0.13
ALG   0.23  0.35     |           0.43     |     0.43
CALC  0.73  0.66     |     0.23  0.95     |           0.97
STAT  0.41  0.63     |           0.75     |    -0.13  0.77
```



Which rotation?

The choice of rotation is a highly contentious topic in factor analysis.

If it scientifically makes sense for the factors to be uncorrelated, then use an orthogonal rotation (varimax)

If this isn't reasonable use an oblique rotation (oblimin)

(Note that, this is essentially a nonparametric vs. parametric trade-off)

How to choose K (the number of factors)?

When selecting the number of factors, researchers attempt to balance parsimony (only a few factors) and plausibility (enough of the original correlation structure is accounted for)

Including too few factors is known as underfactoring. This leads to many problems, essentially related to confounding

Including too many variables is known as overfactoring. This isn't as serious, as the factor loadings on these extra factors should be small.

How to choose K (the number of factors)?

There are many possible ways to choose the number of factors

- Scree plot (tends to pick too many factors)
- Kaiser criterion (criticized as subjective)
- Cumulative variance explained
- Parallel analysis and non-graphical versions of the scree plot
- Using (non-statistical) theory to inform choice
- many others...

Scree plot

We plot the eigenvalues of the correlation matrix, decending from largest to smallest.

We choose the number of factors to be at the ‘elbow’ or ‘kink.’ That is, when the eigenvalues go from decreasing rapidly to slowly.

We can get this information easy enough

```
ev    = eigen(Xcorr)
Kgrid = 1:p
plot(Kgrid,ev,xlab='number of factors', ylab='eigenvals')
```

Scree results on grades data

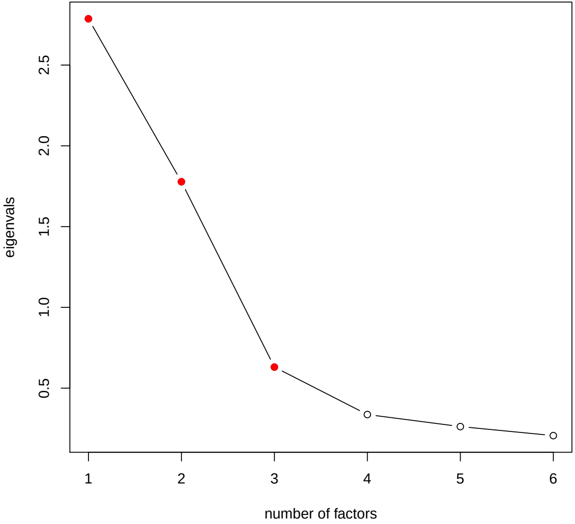


Figure: Choose 3 factors

Kaiser criterion

Alternatively, we can just look at the magnitude of the eigenvalues.

As this is a correlation matrix, the diagonal entries are all 1

Also, it is a fact that if $(\lambda_j)_{j=1}^{p}$ are the eigenvalues ($p = 6$ in our example), then

$$\sum_{j=1}^{p} \lambda_j = \mathrm{trace}(\texttt{cor(X)}) = p$$

where *trace* just sums the diagonal entries.

If there is no factor structure, we might expect that all eigenvalues are about 1 (which is the average)

Therefore, we can keep the factors that correspond to eigenvalues greater than 1

This is known as either the Kaiser criterion or Kaiser's criterion

Kaiser's criterion

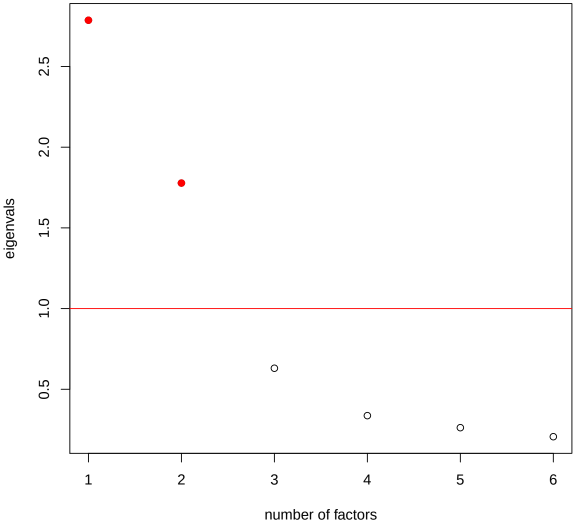


Figure: Choose 2 factors

Cumulative variance explained

A very commonly used technique, at least in psychometrics

The idea is just to pick a threshold (very often 60%), and keep all the factors needed to explain that much variance

This is commonly done be looking at the cumulative sum of the eigenvalues:

```
ev = eigen(Xcorr)
ev$val/p
> ev$val/p
[1] 0.46445471 0.29629698 0.10507729 0.05610791
  0.04368477 0.03437835
> cumsum(ev$val/p)
[1] 0.4644547 0.7607517 0.8658290 0.9219369
  0.9656217 1.0000000
```

Pick 2 factors.

(Horn's) parallel analysis

A Monte-Carlo based simulation method that compares the observed eigenvalues with those obtained from uncorrelated normal variables.

A factor or component is retained if the associated eigenvalue is bigger than some quantile of this null distribution

Parallel analysis is regularly recommended (though it is far from a consensus)

It is somewhat reminiscent of the 'gap statistic' from cluster analysis

```
library(nFactors)
ap = parallel(subject = nrow(X),
var = ncol(X), rep=100,cent=0.05)
# parallel: This makes a correlation matrices from null dist
ns = nScree(ev$values,ap$eigen$qevpea)
plotnScree(ns)
```

(Horn's) parallel analysis

If we type

```
ns = nScree(ev$values,ap$eigen$qevpea)
ns
> ns
  noc naf nparallel nkaiser
    2   2         2       2
```

The acceleration factor (AF) corresponds to a numerical solution to the elbow of the scree plot (that is the second derivative)

The optimal coordinates (OC) corresponds to an extrapolation of the preceeding eigenvalues by a regression line between the eigenvalue coordinates and the last eigenvalue coordinates.

Parallel analysis

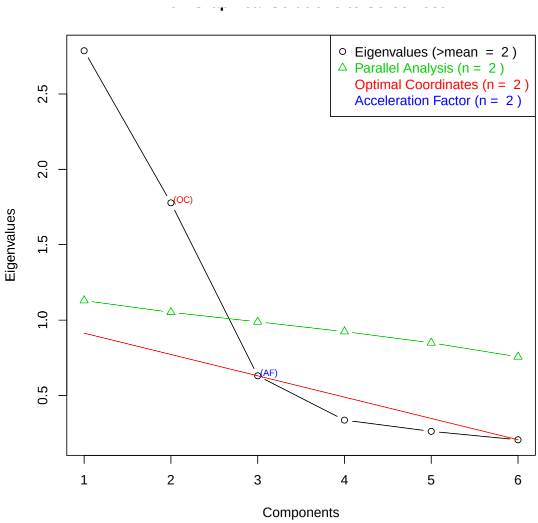


Figure: Note: there is an error in the code for this package for computing AF. They ignore the fact that it is undefined for 1 factor. So, it should say $n = 3$ for AF.