

Dimension reduction through factor analysis, principal components analysis and cluster analysis

William Revelle

Department of Psychology
Northwestern University
Evanston, Illinois USA



NORTHWESTERN
UNIVERSITY

April, 2025

Outline

Preliminaries

Models

Various models

The basic concepts

An example correlation matrix

PCA vs. EFA

Principal Components: An observed Variable Model

Factor Analysis: A Latent Variable Model

Goodness of fit

Maximum Likelihood and its alternatives

Factor Score indeterminacy

Rotations and Transformations

Oblique Transformations

The number of factors/components problem

Simulated data

Another simulation – of a circumplex structure

Real data – Ability tests

Factor diagrams

Introduction

1. Ockham's razor and data reduction
2. Factor analysis – several examples
 - Data from a correlation matrix
 - Simulated 2 factor data
 - Real data – Ability tests
 - Raw data
 - Simulated 2 factor data
 - Real data – 5 Personality dimensions
3. Principal Components analysis
4. Cluster analysis
5. Multidimensional Scaling

Models of data

(MacCallum, 2004) “A factor analysis model is not an exact representation of real-world phenomena.

Always wrong to some degree, even in population.

At best, model is an approximation of real world.”

Box (1979): “Models, of course, are never true, but fortunately it is only necessary that they be useful. For this it is usually needful only that they not be grossly wrong.”

Tukey (1961): “In a single sentence, the moral is: Admit that complexity always increases, first from the model you fit to the data, thence to the model you use to think and plan about the experiment and its analysis, and thence to the true situation.”

(From MacCallum, 2004); <http://www.fa100.info/maccallum2.pdf>

Observed Variables

X

X_1

X_2

X_3

X_4

X_5

X_6

Y

Y_1

Y_2

Y_3

Y_4

Y_5

Y_6

Latent Variables

ξ

η

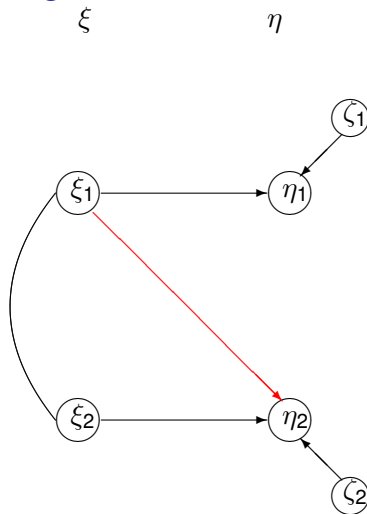
ξ_1

η_1

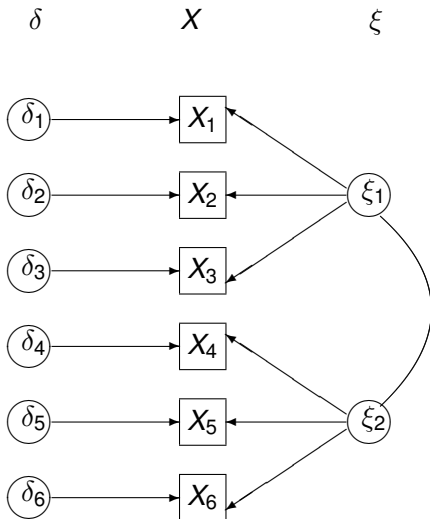
ξ_2

η_2

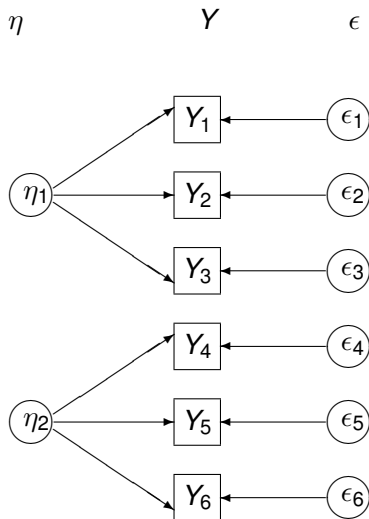
Theory: A regression model of latent variables



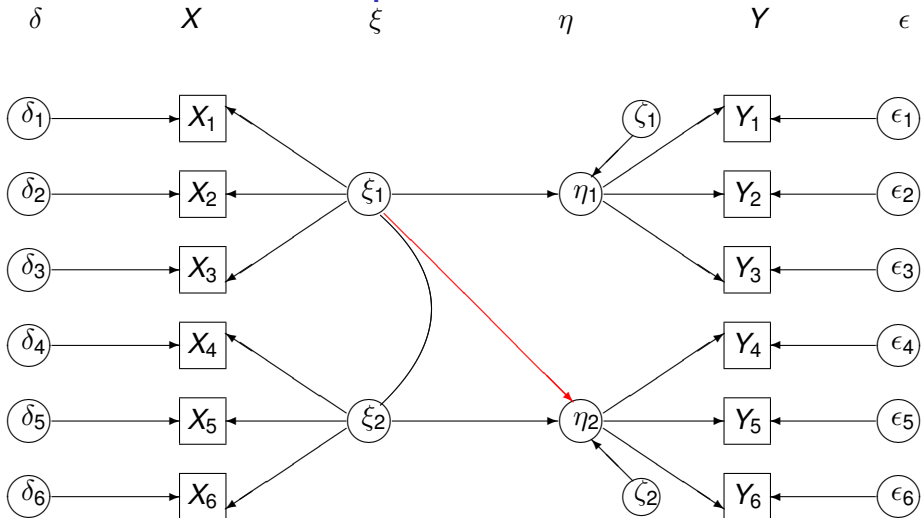
A measurement model for X



A measurement model for Y



A complete structural model



Various measurement models

1. Observed variables models

- Singular Value Decomposition
- Eigen Value – Eigen Vector decomposition
- Principal Components
- First k principal components as an approximation

2. Latent variable models

- Factor analysis
 - Exploratory models – EFA – derivation of theory
 - Confirmatory model – CFA – test of theory

3. Interpretation of models (for EFA)

- Choosing the appropriate number of components/factors
- Transforming/rotating towards interpretable structures

Singular Value Decomposition of the data matrix

Consider the matrix $\mathbf{X}$ of n deviation scores for N subjects, where each element, x_{ij} , represents the responses of the i^{th} individual to the j^{th} item or test. For simplicity, let the x_{ij} scores in each column be deviations from the mean for that column (i.e., they are column centered, perhaps by using `scale`). Let the number of variables be n . Then the `svd` function will find the *Singular Value Decomposition* of $\mathbf{X}$ which allows us to express $\mathbf{X}$ as the product of three orthogonal matrices:

$$_N\mathbf{X}_n = _N\mathbf{U}_{nn}\mathbf{D}_{nn}\mathbf{V}'_n$$

where $\mathbf{D}$ is a diagonal matrix of the *singular values* and the $\mathbf{U}$ and $\mathbf{V}$ matrices are matrices of the *singular vectors*. Although descriptive of the data, what is *meaning* of these vectors?

Decomposition (models) of Correlation and Covariance matrices

With $\mathbf{X}$ defined as before, the covariance matrix, $\mathbf{Cov}$, is

$$\mathbf{Cov} = N^{-1} \mathbf{X}' \mathbf{X}$$

and the standard deviations are

$$\mathbf{sd} = \sqrt{\text{diag}(\mathbf{Cov})}.$$

Let the matrix $\mathbf{I}_{\text{sd}}$ be a diagonal matrix with elements $= \frac{1}{\text{sd}_i}$, then the correlation matrix $\mathbf{R}$ is

$$\mathbf{R} = \mathbf{I}_{\text{sd}} \mathbf{Cov} \mathbf{I}_{\text{sd}}.$$

The problem is how to approximate the matrix, $\mathbf{R}$ of rank n , with a matrix of lower rank? The solution to this problem may be seen if we think about how to create a model matrix to approximate $\mathbf{R}$.

Factor Analysis/Components Analysis/Cluster Analysis

1. Data simplification and Ockham's Razor: "do not multiply entities beyond necessity"
2. Can we describe a data set with a simpler representation of the data.
3. Is it possible to combine subjects and or variables that are redundant?
4. Or almost redundant (without losing very much information)
5. This is a problem in projective geometry. Can we project from a high dimensional space into a lower order space.

An example correlation matrix

Consider the following correlation matrix

	V1	V2	V3	V4	V5	V6
V1	1.00	0.72	0.63	0.54	0.45	0.36
V2	0.72	1.00	0.56	0.48	0.40	0.32
V3	0.63	0.56	1.00	0.42	0.35	0.28
V4	0.54	0.48	0.42	1.00	0.30	0.24
V5	0.45	0.40	0.35	0.30	1.00	0.20
V6	0.36	0.32	0.28	0.24	0.20	1.00

Is it possible to model these 36 correlations and variances with fewer terms? Yes, of course. The diagonal elements are all 1 and the off diagonal elements are symmetric. Thus, we have $n * (n - 1)$ correlations we want to model.

Eigen vector decomposition

Given a $n \times n$ matrix $\mathbf{R}$, each eigenvector, $\mathbf{x}_i$, solves the equation

$$\mathbf{x}_i \mathbf{R} = \lambda_i \mathbf{x}_i$$

and the set of n eigenvectors are solutions to the equation

$$\mathbf{X} \mathbf{R} = \mathbf{X} \boldsymbol{\lambda}$$

where $\mathbf{X}$ is a matrix of orthogonal eigenvectors and $\boldsymbol{\lambda}$ is a diagonal matrix of the the eigenvalues, λ_i . Then

$$\mathbf{x}_i \mathbf{R} - \lambda_i \mathbf{x}_i \mathbf{I} = 0 \iff \mathbf{x}_i (\mathbf{R} - \lambda_i \mathbf{I}) = 0$$

Finding the eigenvectors and eigenvalues is computationally tedious, but may be done using the `eigen` function. That the vectors making up $\mathbf{X}$ are orthogonal means that

$$\mathbf{X} \mathbf{X}' = \mathbf{I}$$

and because they form the *basis space* for $\mathbf{R}$ that

$$\mathbf{R} = \mathbf{X} \boldsymbol{\lambda} \mathbf{X}'.$$

Consider the eigen value solution for the example correlation matrix.

```
> e <- eigen(R)
> print(e, digits=2)
```

\$values

```
[1] 3.16 0.82 0.72 0.59 0.44 0.26
```

\$vectors

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]
[1,]	-0.50	-0.061	0.092	0.14	0.238	0.816
[2,]	-0.47	-0.074	0.121	0.21	0.657	-0.533
[3,]	-0.43	-0.096	0.182	0.53	-0.675	-0.184
[4,]	-0.39	-0.142	0.414	-0.78	-0.201	-0.104
[5,]	-0.34	-0.299	-0.860	-0.20	-0.108	-0.067
[6,]	-0.28	0.934	-0.178	-0.10	-0.067	-0.045

```
> round(e$vectors %*% t(e$vectors),2) #the eigen vectors are orthogonal
```

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]
[1,]	1	0	0	0	0	0
[2,]	0	1	0	0	0	0
[3,]	0	0	1	0	0	0
[4,]	0	0	0	1	0	0
[5,]	0	0	0	0	1	0
[6,]	0	0	0	0	0	1

Eigen Value decomposition and recreation of the original matrix

Find the eigen values (λ) and eigen vectors ($\mathbf{V}_i$).

```
> e <- eigen(R)
> print(e, digits=2)
```

```
$values
[1] 3.16 0.82 0.72 0.59 0.44 0.26
```

```
$vectors
      [,1] [,2] [,3] [,4] [,5] [,6]
[1,] -0.50 -0.061 0.092 0.14 0.238 0.816
[2,] -0.47 -0.074 0.121 0.21 0.657 -0.533
[3,] -0.43 -0.096 0.182 0.53 -0.675 -0.184
[4,] -0.39 -0.142 0.414 -0.78 -0.201 -0.104
[5,] -0.34 -0.299 -0.860 -0.20 -0.108 -0.067
[6,] -0.28 0.934 -0.178 -0.10 -0.067 -0.045
```

#the eigen vectors are orthogonal

```
> round(e$vectors %*% t(e$vectors),2)
      [,1] [,2] [,3] [,4] [,5] [,6]
[1,] 1 0 0 0 0 0
[2,] 0 1 0 0 0 0
[3,] 0 0 1 0 0 0
[4,] 0 0 0 1 0 0
[5,] 0 0 0 0 1 0
[6,] 0 0 0 0 0 1
```

The eigen vectors and values recreate the observed correlations.

$$\mathbf{R} = \mathbf{V}\boldsymbol{\lambda}\mathbf{V}'$$

```
> round(e$vectors %*% diag(e$values) %*% t(e$vectors),2)
```

```
      [,1] [,2] [,3] [,4] [,5] [,6]
[1,] 1.00 0.72 0.63 0.54 0.45 0.36
[2,] 0.72 1.00 0.56 0.48 0.40 0.32
[3,] 0.63 0.56 1.00 0.42 0.35 0.28
[4,] 0.54 0.48 0.42 1.00 0.30 0.24
[5,] 0.45 0.40 0.35 0.30 1.00 0.20
[6,] 0.36 0.32 0.28 0.24 0.20 1.00
```

The eigen values reflect the scale, the vectors the structure

Consider the original data and solution

```
> R
> e <- eigen(R)
> print(e, digits=2)
```

	V1	V2	V3	V4	V5	V6
V1	1.00	0.72	0.63	0.54	0.45	0.36
V2	0.72	1.00	0.56	0.48	0.40	0.32
V3	0.63	0.56	1.00	0.42	0.35	0.28
V4	0.54	0.48	0.42	1.00	0.30	0.24
V5	0.45	0.40	0.35	0.30	1.00	0.20
V6	0.36	0.32	0.28	0.24	0.20	1.00

```
$values
[1] 3.16 0.82 0.72 0.59 0.44 0.26
```

```
$vectors
      [,1] [,2] [,3] [,4] [,5] [,6]
[1,] -0.50 -0.061 0.092 0.14 0.238 0.816
[2,] -0.47 -0.074 0.121 0.21 0.657 -0.533
[3,] -0.43 -0.096 0.182 0.53 -0.675 -0.184
[4,] -0.39 -0.142 0.414 -0.78 -0.201 -0.104
[5,] -0.34 -0.299 -0.860 -0.20 -0.108 -0.067
[6,] -0.28 0.934 -0.178 -0.10 -0.067 -0.045
```

Consider if all the correlations are divided by 2.

```
> R.5 <- as.matrix(R/2)
> diag(R.5) <- 1
> R.5
> e.5 <- eigen(R.5)
> print(e.5,2)
```

	V1	V2	V3	V4	V5	V6
V1	1.000	0.36	0.315	0.27	0.225	0.18
V2	0.360	1.00	0.280	0.24	0.200	0.16
V3	0.315	0.28	1.000	0.21	0.175	0.14
V4	0.270	0.24	0.210	1.00	0.150	0.12
V5	0.225	0.20	0.175	0.15	1.000	0.10
V6	0.180	0.16	0.140	0.12	0.100	1.00

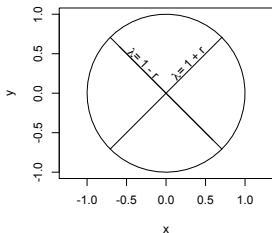
```
$values
[1] 2.08 0.91 0.86 0.80 0.72 0.63
```

```
$vectors
      [,1] [,2] [,3] [,4] [,5] [,6]
[1,] 0.50 -0.061 0.092 0.14 0.238 0.816
[2,] 0.47 -0.074 0.121 0.21 0.657 -0.533
[3,] 0.43 -0.096 0.182 0.53 -0.675 -0.184
[4,] 0.39 -0.142 0.414 -0.78 -0.201 -0.104
[5,] 0.34 -0.299 -0.860 -0.20 -0.108 -0.067
[6,] 0.28 0.934 -0.178 -0.10 -0.067 -0.045
```

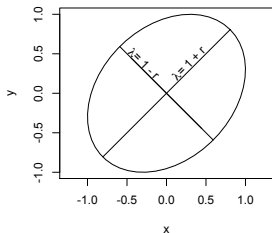
Note that the signs are arbitrary.

Eigen vectors of a 2 x 2 correlation matrix

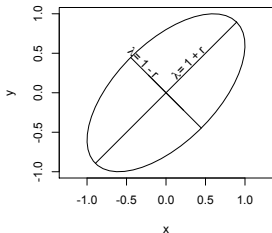
$r = 0$



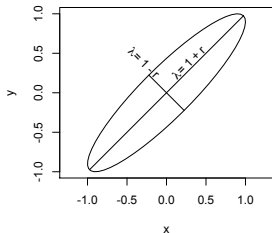
$r = 0.3$



$r = 0.6$



$r = 0.9$



Although the length (eigen values) of the axes differ, their orientation (eigen vectors) are the same.

```
> r2 <- matrix(c(1,.6,.6,1),2,2)
> print(eigen(r2),2)
```

\$values

```
[1] 1.6 0.4
```

\$vectors

```
      [,1] [,2]
[1,] 0.71 -0.71
[2,] 0.71  0.71
```

From eigen vectors to Principal Components

1. For n variables, there are n eigen vectors
 - There is no parsimony in thinking of the eigen vectors
 - Except that the vectors provide the orthogonal basis for the variables
2. Principal components are formed from the eigen vectors and eigen values
 - $\mathbf{R} = \mathbf{V}\lambda\mathbf{V}' = \mathbf{C}\mathbf{C}'$
 - $\mathbf{C} = \mathbf{V} \sqrt{\lambda}$
3. But there will still be as many Principal Components as variables, so what is the point?
4. Take just the first k Principal Components and see how well this reduced model fits the data.

The first principal component.

```
> pc1 <- principal(R,1)
> round(e$vectors[,1] *
      sqrt(e$values[1]),2)
```

```
-0.88 -0.83 -0.77 -0.69 -0.60 -0.50
```

```
> pc1
```

```
Uniquenesses:
```

```
      V1      V2      V3      V4      V5
```

```
V6
```

```
0.220 0.307 0.408 0.519 0.635 0.748
```

```
Loadings:
```

```
      PC1
```

```
V1 0.88
```

```
V2 0.83
```

```
V3 0.77
```

```
V4 0.69
```

```
V5 0.60
```

```
V6 0.50
```

```
      PC1
```

```
SS loadings      3.142
```

```
Proportion Var 0.524
```

```
#show the model
```

```
> round(pc1$loadings %>% t(pc1$loadings),2)
```

	V1	V2	V3	V4	V5	V6
V1	0.77	0.73	0.68	0.61	0.53	0.44
V2	0.73	0.69	0.64	0.57	0.50	0.42
V3	0.68	0.64	0.59	0.53	0.46	0.38
V4	0.61	0.57	0.53	0.48	0.41	0.34
V5	0.53	0.50	0.46	0.41	0.36	0.30
V6	0.44	0.42	0.38	0.34	0.30	0.25

```
#find the residuals
```

```
> Rresid <-
```

```
R - pc1$loadings %>% t(pc1$loadings)
```

```
> round(Rresid,2)
```

	V1	V2	V3	V4	V5	V6
V1	0.23	-0.01	-0.05	-0.07	-0.08	-0.08
V2	-0.01	0.31	-0.08	-0.09	-0.10	-0.09
V3	-0.05	-0.08	0.41	-0.11	-0.11	-0.10
V4	-0.07	-0.09	-0.11	0.52	-0.11	-0.10
V5	-0.08	-0.10	-0.11	-0.11	0.64	-0.10
V6	-0.08	-0.09	-0.10	-0.10	-0.10	0.75

The model fits pretty well, except that the diagonal is

Try 2 and 3 principal components

```
> p2 <- principal(R, 2, rotate = "none")
> p2
> resid(p2)
```

Principal Components Analysis

Call: principal(r = R, nfactors = 2, rotate = "none")

Standardized loadings (pattern **matrix**)

	PC1	PC2	h2	u2
V1	0.88	-0.06	0.78	0.217
V2	0.83	-0.07	0.70	0.302
V3	0.77	-0.09	0.60	0.400
V4	0.69	-0.13	0.50	0.502
V5	0.60	-0.27	0.44	0.561
V6	0.50	0.85	0.97	0.031

	PC1	PC2
SS loadings	3.16	0.82
Proportion Var	0.53	0.14
Cumulative Var	0.53	0.66

Fit based upon **off** diagonal values = 0.95

	V1	V2	V3	V4	V5	V6
V1	0.22					
V2	-0.02	0.30				
V3	-0.05	-0.09	0.40			
V4	-0.08	-0.11	-0.13	0.50		
V5	-0.10	-0.12	-0.14	-0.15	0.56	
V6	-0.04	-0.04	-0.03	0.00	0.13	0.03

```
> p3 <- principal(R, 3, rotate = "none")
> resid(p3)
```

Principal Components Analysis

Call: principal(r = R, nfactors = 3, rotate = "none")

Standardized loadings (pattern **matrix**)

	PC1	PC2	PC3	h2	u2
V1	0.88	-0.06	-0.08	0.79	0.2108
V2	0.83	-0.07	-0.10	0.71	0.2917
V3	0.77	-0.09	-0.15	0.62	0.3761
V4	0.69	-0.13	-0.35	0.62	0.3789
V5	0.60	-0.27	0.73	0.97	0.0292
V6	0.50	0.85	0.15	0.99	0.0084

	PC1	PC2	PC3
SS loadings	3.16	0.82	0.72
Proportion Var	0.53	0.14	0.12
Cumulative Var	0.53	0.66	0.78

Fit based upon **off** diagonal values = 0.97

	V1	V2	V3	V4	V5	V6
V1	0.21					
V2	-0.03	0.29				
V3	-0.07	-0.10	0.38			
V4	-0.11	-0.14	-0.18	0.38		
V5	-0.04	-0.05	-0.03	0.10	0.03	
V6	-0.02	-0.03	-0.01	0.05	0.02	0.01

Consider the following matrix

Correlations between 6 variables						
Variable	V1	V2	V3	V4	V5	V6
V1	1.00	0.72	0.63	0.54	0.45	0.36
V2	0.72	1.00	0.56	0.48	0.40	0.32
V3	0.63	0.56	1.00	0.42	0.35	0.28
V4	0.54	0.48	0.42	1.00	0.30	0.24
V5	0.45	0.40	0.35	0.30	1.00	0.20
V6	0.36	0.32	0.28	0.24	0.20	1.00

Can we represent this in a simpler way?

$$R = FF' + U^2$$

or

$$R = CC'$$

Representing a correlation matrix with factors or components

Correlations between 6 variables

Variable	V1	V2	V3	V4	V5	V6
V1	1.00	0.72	0.63	0.54	0.45	0.36
V2	0.72	1.00	0.56	0.48	0.40	0.32
V3	0.63	0.56	1.00	0.42	0.35	0.28
V4	0.54	0.48	0.42	1.00	0.30	0.24
V5	0.45	0.40	0.35	0.30	1.00	0.20
V6	0.36	0.32	0.28	0.24	0.20	1.00

Table: $R = FF' + U^2$

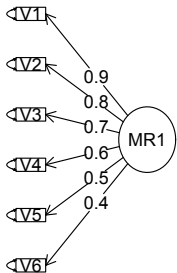
Variable	loading
V1	0.9
V2	0.8
V3	0.7
V4	0.6
V5	0.5
V6	0.4

Table: $R = CC'$

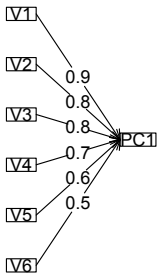
Variable	
V1	0.88
V2	0.83
V3	0.77
V4	0.69
V5	0.60
V6	0.50

The factor model versus the components model

Factor model



Components model



Factors vs. components

Originally developed by [Spearman \(1904\)](#) for the case of one common factor, and then later generalized by [Thurstone \(1947\)](#) and others to the case of multiple factors, factor analysis is probably the most frequently used and sometimes the most controversial psychometric procedure. The factor model, although seemingly very similar to the components model, is in fact very different. For rather than having components as linear sums of variables, in the factor model the variables are themselves linear sums of the unknown factors. That is, while components can be solved for by doing an *eigenvalue* or *singular value decomposition*, factors are estimated as best fitting solutions ([Eckart and Young, 1936](#); [Householder and Young, 1938](#)), normally through iterative methods ([Jöreskog, 1978](#); [Lawley and Maxwell, 1963](#)). [Cattell \(1965\)](#) referred to components analysis as a closed model and factor analysis as an open model, in that by explaining just the common variance, there was still more variance to explain.

Iterative principal axes factor analysis

Principal components represents a $n * n$ matrix in terms of the first k components. It attempts to reproduce all of the **R** matrix. *Factor analysis* on the other hand, attempts to model just the common part of the matrix, which means all of the off-diagonal elements and the common part of the diagonal (the *communalities*). The non-common part, the *uniquenesses*, are simply that which is left over. An easy to understand procedure is *principal axes* factor analysis. This is similar to principal components, except that it is done with a reduced matrix where the diagonals are the communalities. The communalities can either be specified a priori, estimated by such procedures as multiple linear regression, or found by iteratively doing an eigenvalue decomposition and repeatedly replacing the original 1s on the diagonal with the value of $1 - u^2$ where

$$\mathbf{U}^2 = \text{diag}(\mathbf{R} - \mathbf{F}\mathbf{F}').$$

Principal axes as eigen values of a reduced matrix

That is, starting with the original correlation or covariance matrix, $\mathbf{R}$, find the k largest principal components, reproduce the matrix using those principal components. Find the resulting residual matrix, $\mathbf{R}^*$ and uniqueness matrix, $\mathbf{U}^2$ by

$$\mathbf{R}^* = \mathbf{R} - \mathbf{F}\mathbf{F}' \quad (1)$$

$$\mathbf{U}^2 = \text{diag}(\mathbf{R}^*)$$

and then, for iteration i , find $\mathbf{R}_i$ by replacing the diagonal of the original $\mathbf{R}$ matrix with $1 - \text{diag}(\mathbf{U}^2)$ found on the previous step. Repeat this process until the change from one iteration to the next is arbitrarily small.

Comparing 1 with 5 iterations

```
> f1 <- fa(R,1,fm='pa',max.iter=1)
> f1
> resid(f1)
```

Factor Analysis using method = pa

Call: fa(r = R, nfactors = 1, max.iter = 1, fm = "pa")

Standardized loadings (pattern matrix)

	PA1	h2	u2
V1	0.86	0.74	0.26
V2	0.79	0.62	0.38
V3	0.70	0.48	0.52
V4	0.60	0.36	0.64
V5	0.50	0.25	0.75
V6	0.40	0.16	0.84

	PA1
SS loadings	2.62
Proportion Var	0.44

	V1	V2	V3	V4	V5	V6
V1	0.26					
V2	0.04	0.38				
V3	0.03	0.01	0.52			
V4	0.02	0.01	0.00	0.64		
V5	0.02	0.00	0.00	0.00	0.75	
V6	0.01	0.00	0.00	0.00	0.00	0.84

```
> f1 <- fa(R,1,fm='pa',max.iter=5)
> f1
> resid(f1)
```

Factor Analysis using method = pa

Call: fa(r = R, nfactors = 1, max.iter = 5, fm = "pa")

Standardized loadings (pattern matrix)

	PA1	h2	u2
V1	0.9	0.81	0.19
V2	0.8	0.64	0.36
V3	0.7	0.49	0.51
V4	0.6	0.36	0.64
V5	0.5	0.25	0.75
V6	0.4	0.16	0.84

	PA1
SS loadings	2.71
Proportion Var	0.45

	V1	V2	V3	V4	V5	V6
V1	0.19					
V2	0.00	0.36				
V3	0.00	0.00	0.51			
V4	0.00	0.00	0.00	0.64		
V5	0.00	0.00	0.00	0.00	0.75	
V6	0.00	0.00	0.00	0.00	0.00	0.84

SMCs as initial communality estimates

Rather than starting with initial communality estimates of 1, the process can be started with other estimates of the communality. A conventional starting point is the lower bound estimate of the communalities, the *squared multiple correlation* or *SMC* (Roff, 1936).

The concept here is that a variable's communality must be at least as great as the amount of its variance that can be predicted by all of the other variables. The squared multiple correlations of each variable with the remaining variables are the diagonal elements of

$$\mathbf{I} - (\text{diag}(\mathbf{R}^{-1}))^{-1}$$

and thus a starting estimate for $\mathbf{R}_0$ would be $\mathbf{R} - (\text{diag}(\mathbf{R}^{-1}))^{-1}$.

Goodness of fit—simple estimates

At least three indices of goodness of fit of the principal factors model can be considered: One compares the sum of squared residuals to the sum of the squares of the original values:

$$GF_{total} = 1 - \frac{\mathbf{1R}^{*2}\mathbf{1}'}{\mathbf{1R}^2\mathbf{1}'}$$

The second does the same, but does not consider the diagonal of **R**

$$GF_{offdiagonal} = 1 - \frac{\sum_{i \neq j} r_{ij}^{*2}}{\sum_{i \neq j} r_{ij}^2} = 1 - \frac{\mathbf{1R}^{*2}\mathbf{1}' - tr(\mathbf{R}^{*2})}{\mathbf{1R}^2\mathbf{1}' - tr(\mathbf{R}^2)}$$

Finally, a χ^2 test of the size of the residuals (r_*) simply sums all the squared residuals and multiplies by the number of observations:

$$\chi^2 = \sum_{i < j} r_{*ij}^2 (N - 1)$$

with $p * (p-1)/2$ degrees of freedom.

OLS

The fundamental factor equation (Equation 1) may be viewed as set of simultaneous equations which may be solved several different ways: *ordinary least squares*, *generalized least squares*, and *maximum likelihood*. Ordinary least squares (*OLS*) or *unweighted least squares* (*ULS*) minimizes the sum of the squared residuals when modeling the sample correlation or covariance matrix, **S**, with $\Sigma = \mathbf{FF}' + \mathbf{U}^2$

$$E = \frac{1}{2} \text{tr}(\mathbf{S} - \Sigma)^2 \quad (2)$$

where the *trace*, tr , of a matrix is the sum of the diagonal elements and the division by two reflects the symmetry of the **S** matrix.

MLE

Equation 2 can be generalized to weight the residuals ($\mathbf{S} - \Sigma$) by the inverse of the sample matrix, $\mathbf{S}$, and thus to minimize

$$E = \frac{1}{2} \text{tr}((\mathbf{S} - \Sigma)\mathbf{S}^{-1})^2 = \frac{1}{2} \text{tr}(\mathbf{I} - \Sigma\mathbf{S}^{-1})^2. \quad (3)$$

This is known as *generalized least squares (GLS)* or *weighted least squares (WLS)*. Similarly, if the residuals are weighted by the inverse of the model, Σ , minimizing

$$E = \frac{1}{2} \text{tr}((\mathbf{S} - \Sigma)\Sigma^{-1})^2 = \frac{1}{2} \text{tr}(\mathbf{S}\Sigma^{-1} - \mathbf{I})^2 \quad (4)$$

will result in a model that maximizes the likelihood of the data. This procedure, *maximum likelihood estimation (MLE)* is also seen as finding the minimum of

$$E = \frac{1}{2} (\text{tr}(\Sigma^{-1}\mathbf{S}) - \ln |\Sigma^{-1}\mathbf{S}| - p) \quad (5)$$

where p is the number of variables (Jöreskog, 1978). Perhaps a helpful intuitive explanation of Equation 8 is that if the model is

Maximum Likelihood Estimation

(MLE) is also seen as finding the minimum of

$$E = \frac{1}{2} (tr(\Sigma^{-1}\mathbf{S}) - \ln |\Sigma^{-1}\mathbf{S}| - p) \quad (8)$$

where p is the number of variables ([Jöreskog, 1978](#)). Perhaps a helpful intuitive explanation of Equation 8 is that if the model is correct, then $\Sigma = \mathbf{S}$ and thus $\Sigma^{-1}\mathbf{S} = \mathbf{I}$. The trace of an identity matrix of rank p is p , and the logarithm of $|\mathbf{I}|$ is 0. Thus, the value of E if the model has perfect fit is 0. With the assumption of multivariate normality of the residuals, and for large samples, a χ^2 statistic can be estimated for a model with p variables and f factors:

$$\chi^2 = (tr(\Sigma^{-1}\mathbf{S}) - \ln |\Sigma^{-1}\mathbf{S}| - p) (N - 1 - (2p + 5)/6 - (2f)/3). \quad (9)$$

This χ^2 has degrees of freedom:

$$df = p * (p - 1)/2 - p * f + f * (f - 1)/2. \quad (10)$$

That is, the number of lower off-diagonal correlations - the number of unconstrained loadings ([Lawley and Maxwell 1962](#))

Minimum Residual Factor Analysis

The previous factor analysis procedures attempt to optimize the fit of the model matrix (Σ) to the correlation or covariance matrix ($\mathbf{S}$). The diagonal of the matrix is treated as mixture of common variance and unique variance and the problem becomes one of estimating the common variance (the *communality* of each variable). An alternative is to ignore the diagonal and to find that model which minimizes the squared residuals of the off diagonal elements. This is done in the `fa` function using the “minres” option by finding the solution that minimizes

$$\frac{1}{2} \mathbf{1}' ((\mathbf{S} - \mathbf{I}) - (\Sigma - \text{tr}(\Sigma)) \mathbf{1}) \mathbf{1}'. \quad (11)$$

The advantage of the *minres* solution is that it does not require finding the inverse of either the original correlation matrix (as do *GLS* and *WLS*) nor of the model matrix (as does *MLE*, and thus can be performed on non-positive definite matrices or matrices that are not invertible.

Solutions with more than 1 factor or component

Nothing in the previous algebra restricted the dimensionality of the **F** matrix or **C** matrix to be one column. That is, why limit ourselves to a one dimensional solution? Consider the following correlation matrix (constructed by creating a factor matrix and then finding its inner product).

```
>F <- matrix(c(.9,.8,.7,rep(0,6),.8,.7,.6),ncol=2) #the model
> rownames(F) <- paste("V",seq(1:6),sep=" ") #add labels
> colnames(F) <- c("F1", "F2")
> R <- F %*% t(F) #create the correlation matrix
> diag(R) <- 1 #adjust the diagonal of the matrix
> R
```

	V1	V2	V3	V4	V5	V6
V1	1.00	0.72	0.63	0.00	0.00	0.00
V2	0.72	1.00	0.56	0.00	0.00	0.00
V3	0.63	0.56	1.00	0.00	0.00	0.00
V4	0.00	0.00	0.00	1.00	0.56	0.48
V5	0.00	0.00	0.00	0.56	1.00	0.42
V6	0.00	0.00	0.00	0.48	0.42	1.00

Try one principal component to this model.

```
> pc1 <- principal(R)
> pc1
```

Principal Components Analysis

Call: principal(r = R)

Standardized loadings (pattern **matrix**)
based upon correlation **matrix**

	PC1	h2	u2
V1	0.90	0.82	0.18
V2	0.88	0.77	0.23
V3	0.83	0.69	0.31
V4	0.00	0.00	1.00
V5	0.00	0.00	1.00
V6	0.00	0.00	1.00

	PC1
SS loadings	2.28
Proportion Var	0.38

Test of the hypothesis that 1 component **is** sufficient.

The degrees of freedom **for** the **null model** are 15
and the objective **function** was 1.96
The degrees of freedom **for** the **model** are 9
and the objective **function** was 0.87

Fit based upon **off** diagonal values = 0.61

The residuals are large for the second set of variables.

```
> resid(pc1)
```

	V1	V2	V3	V4	V5	V6
V1	0.18					
V2	-0.07	0.23				
V3	-0.12	-0.17	0.31			
V4	0.00	0.00	0.00	1.00		
V5	0.00	0.00	0.00	0.56	1.00	
V6	0.00	0.00	0.00	0.48	0.42	1.00

Two Principal Components

```
> pc2 <- principal(R,2)
> pc2
```

Uniquenesses:

	V1	V2	V3	V4	V5	V6
	0.182	0.234	0.309	0.282	0.332	0.409

Loadings:

	PC1	PC2
V1	0.90	
V2	0.88	
V3	0.83	
V4		0.85
V5		0.82
V6		0.77

	PC1	PC2
SS loadings	2.273	1.988
Proportion Var	0.379	0.331
Cumulative Var	0.379	0.710

```
> round(pc2$loadings %>% t(pc2$loadings),2)
```

	V1	V2	V3	V4	V5	V6
V1	0.81	0.79	0.75	0.00	0.00	0.00
V2	0.79	0.77	0.73	0.00	0.00	0.00
V3	0.75	0.73	0.69	0.00	0.00	0.00
V4	0.00	0.00	0.00	0.72	0.70	0.65
V5	0.00	0.00	0.00	0.70	0.67	0.63
V6	0.00	0.00	0.00	0.65	0.63	0.59

```
> Rresid <- R - pc2$loadings %>% t(pc2$loadings)
> round(Rresid,2)
```

	V1	V2	V3	V4	V5	V6
V1	0.19	-0.07	-0.12	0.00	0.00	0.00
V2	-0.07	0.23	-0.17	0.00	0.00	0.00
V3	-0.12	-0.17	0.31	0.00	0.00	0.00
V4	0.00	0.00	0.00	0.28	-0.14	-0.17
V5	0.00	0.00	0.00	-0.14	0.33	-0.21
V6	0.00	0.00	0.00	-0.17	-0.21	0.41

```
> resid(pc2)
```

	V1	V2	V3	V4	V5	V6
V1	0.18					
V2	-0.07	0.23				
V3	-0.12	-0.17	0.31			
V4	0.00	0.00	0.00	0.28		
V5	0.00	0.00	0.00	-0.13	0.33	
V6	0.00	0.00	0.00	-0.17	-0.21	0.41

Try two factors

```
> f2 <- fa(R, 2, rotate = "none")
> f2
```

Factor Analysis using method = minres

Call: fa(r = R, nfactors = 2, rotate = "none")

Standardized loadings (pattern **matrix**) based upon correlation **matrix**

	MR1	MR2	h2	u2
V1	0.9	0.0	0.81	0.19
V2	0.8	0.0	0.64	0.36
V3	0.7	0.0	0.49	0.51
V4	0.0	0.8	0.64	0.36
V5	0.0	0.7	0.49	0.51
V6	0.0	0.6	0.36	0.64

	MR1	MR2
SS loadings	1.94	1.49
Proportion Var	0.32	0.25
Cumulative Var	0.32	0.57

Test of the hypothesis that 2 factors are sufficient.

The degrees of freedom **for** the **null model** are 15
 and the objective **function** was 1.96
 The degrees of freedom **for** the **model** are 4
 and the objective **function** was 0

The root **mean** square of the **residuals** (RMSR) is 0
 The **df** corrected root **mean** square of the **residuals** is 0

Fit based upon **off** diagonal values = 1

```
> resid(f2)
```

	V1	V2	V3	V4	V5	V6
V1	0.19					
V2	0.00	0.36				
V3	0.00	0.00	0.51			
V4	0.00	0.00	0.00	0.36		
V5	0.00	0.00	0.00	0.00	0.51	
V6	0.00	0.00	0.00	0.00	0.00	0.64

Add two more variables (with a factor model)

#the model

```
> f <- matrix(c(.9,.8,.7,rep(0,3),.7,rep(0,4),.8,.7,.6,0,.5),ncol=2)
> rownames(f) <- paste("V",seq(1:8),sep="") #add labels
> colnames(f) <- c("F1", "F2")
> R <- f %*% t(f) #create the correlation matrix
> diag(R) <- 1 #adjust the diagonal of the matrix
> R
```

	V1	V2	V3	V4	V5	V6	V7	V8
V1	1.00	0.72	0.63	0.00	0.00	0.00	0.63	0.00
V2	0.72	1.00	0.56	0.00	0.00	0.00	0.56	0.00
V3	0.63	0.56	1.00	0.00	0.00	0.00	0.49	0.00
V4	0.00	0.00	0.00	1.00	0.56	0.48	0.00	0.40
V5	0.00	0.00	0.00	0.56	1.00	0.42	0.00	0.35
V6	0.00	0.00	0.00	0.48	0.42	1.00	0.00	0.30
V7	0.63	0.56	0.49	0.00	0.00	0.00	1.00	0.00
V8	0.00	0.00	0.00	0.40	0.35	0.30	0.00	1.00

Factors loadings do not change, component loadings do

> **R**

```

      V1  V2  V3  V4  V5  V6  V7  V8
V1 1.00 0.72 0.63 0.00 0.00 0.00 0.63 0.00
V2 0.72 1.00 0.56 0.00 0.00 0.00 0.56 0.00
V3 0.63 0.56 1.00 0.00 0.00 0.00 0.49 0.00
V4 0.00 0.00 0.00 1.00 0.56 0.48 0.00 0.40
V5 0.00 0.00 0.00 0.56 1.00 0.42 0.00 0.35
V6 0.00 0.00 0.00 0.48 0.42 1.00 0.00 0.30
V7 0.63 0.56 0.49 0.00 0.00 0.00 1.00 0.00
V8 0.00 0.00 0.00 0.40 0.35 0.30 0.00 1.00

```

```

> f2 <- factanal(covmat=R, factors=2)
> f2

```

Call:

```
factanal(factors = 2, covmat = R)
```

Uniquenesses:

```

  V1  V2  V3  V4  V5  V6  V7  V8
0.19 0.36 0.51 0.36 0.51 0.64 0.51 0.75

```

Loadings:

```

  Factor1 Factor2

```

```

V1 0.9
V2 0.8
V3 0.7
V4
V5 0.8
V6 0.6
V7 0.7
V8 0.5

```

```

pc2 <- principal(R,2)
pc2

```

Uniquenesses:

```

  V1  V2  V3  V4  V5  V6
V7  V8
0.194 0.271 0.367 0.311 0.379 0.468 0.367 0.575

```

Loadings:

```

  PC1 PC2
V1 0.90
V2 0.85
V3 0.80
V4 0.83
V5 0.79
V6 0.73
V7 0.80
V8 0.65

```

```

          PC1  PC2
SS loadings 2.812 2.268
Proportion Var 0.352 0.284
Cumulative Var 0.352 0.635

```

Factor Scores are estimates and are only correlated with the factors

Given the data matrix, ${}_n\mathbf{X}_v$ with elements X_{ij} , we can find the deviation matrix, ${}_n\mathbf{x}_v$ and the covariance matrix between the observed variables, ${}_v\mathbf{C}_v$. This matrix may be reproduced by the product of the factor matrix and its transpose. See Equation 1 which is reproduced here with subscripts to show the number of observed covariances ($\frac{v(v-1)}{2}$) and the number of parameters to find ($vk - \frac{f*(f-1)}{2}$):

$${}_v\mathbf{C}_v \approx {}_v\mathbf{F}_{f \cdot f}\mathbf{F}'_v + {}_v\mathbf{U}_v^2 \quad (12)$$

where the ${}_v\mathbf{U}_v^2$ values are found by subtracting the diagonal of the modeled covariances from the diagonal of the observed covariances.

We need to find *factor scores*, ${}_n\mathbf{S}_f$, and the uniquenesses, ${}_n\mathbf{U}_v$ such that

$${}_n\mathbf{x}_v = {}_n\mathbf{S}_{f \cdot f}\mathbf{F}'_v + {}_n\mathbf{U}_v = ({}_v\mathbf{F}_{f \cdot f}\mathbf{S}'_n + {}_v\mathbf{U}_n)' \quad (13)$$

and

Factor Scores are estimates: II

Dropping the subscripts to make the equation more legible we find

$$\mathbf{C} = \mathbf{x}'\mathbf{x}/n = \mathbf{F}\mathbf{S}'\mathbf{S}\mathbf{F}'$$

and because the factors are orthogonal, $\mathbf{S}'\mathbf{S} = \mathbf{I}$,

$$\mathbf{C} = \mathbf{x}'\mathbf{x}/n = \mathbf{F}\mathbf{F}'.$$

Unfortunately, the number of free parameters in Equation 13 is $nv + nf + fv$ which, of course, exceeds the number of observed data points nv . That is, we have more unknowns than knowns and there are an infinity of possible solutions. We can see the problem more directly when solving equation 13 for $\hat{\mathbf{S}}$. Post-multiplying each side of the equation by ${}_v\mathbf{F}_f$ leads to

$$({}_n\mathbf{x}_v - {}_n\mathbf{U}_v){}_v\mathbf{F}_f = {}_n\hat{\mathbf{S}}_{f \cdot f}\mathbf{F}'_{v \cdot v}\mathbf{F}_f = {}_n\hat{\mathbf{S}}_{f \cdot f}\mathbf{C}_f$$

and thus

$$\hat{\mathbf{S}} = (\mathbf{x} - \mathbf{U})\mathbf{F}\mathbf{C}^{-1} = \mathbf{x}\mathbf{F}\mathbf{C}^{-1} - \mathbf{U}\mathbf{F}\mathbf{C}^{-1}. \quad (14)$$

Factor Scores are estimates: III

The problem of finding *factor score estimates*, $\hat{\mathbf{S}}$, is that while there is a observable part, $\mathbf{xFC}^{-1}$, of the score $\mathbf{S}$, there is also an unobservable part, $\mathbf{UFC}^{-1}$. Unless the communalities are one and therefore the uniquenesses are zero (that is, unless we are doing a *components analysis*, the factor scores are indeterminate, although a best *estimate* of the factor scores (in terms of a least squares regression) will be

$$\hat{\mathbf{S}} = \mathbf{xFC}^{-1} = \mathbf{xW}$$

where

$$\mathbf{W} = \mathbf{FC}^{-1} \quad (15)$$

is just a matrix of the β weights for estimating factor scores from the observed variables and R^2 between these estimates and factors is

$$R^2 = \text{diag}(\mathbf{WF}). \quad (16)$$

fa provides *factor score estimates* as well as measures of the *indeterminacy* of the solution.

Factor Scores are estimates: Regression weights and idealized weights

Unfortunately, even for uncorrelated factors, the regression based weights found by Equation 15 will not necessarily produce uncorrelated factor score estimates. Nor, if the factors are correlated, will the factor scores have the same correlations.

Gorsuch (1983) and Grice (2001) review several alternative ways to estimate factor scores, some of which will preserve orthogonality if the factors are orthogonal, others that will preserve the correlations between factors.

The regression solution (Equation 15) will produce factor score estimates that are most correlated with the factors, but will not preserve the factor intercorrelations (or, in the case of orthogonal factors, their orthogonality).

Harman (1976) proposed to weight the factor loadings based upon *idealized variables* by finding the inverse of the inner product of the factor loadings:

$${}_k\mathbf{W}_v = ({}_k\mathbf{F}_v' \mathbf{F}_k)^{-1} {}_k\mathbf{F}_v. \quad (17)$$

Bartlett and tenBerge weights

A somewhat different least squares procedure was proposed by [Bartlett \(1937\)](#) to minimize the contribution of the unique factors:

$$\mathbf{W} = \mathbf{U}^{-2}\mathbf{F}(\mathbf{F}'\mathbf{U}^{-2}\mathbf{F})^{-1}. \quad (18)$$

A variation of Equation 18 proposed by [Anderson and Rubin \(1956\)](#) requires that the factors be orthogonal, and preserves this orthogonality:

$$\mathbf{W} = \mathbf{U}^{-2}\mathbf{F}(\mathbf{F}'\mathbf{U}^{-2}\mathbf{C}\mathbf{U}^{-2}\mathbf{F})^{-1/2}. \quad (19)$$

This solution was generalized to the oblique factors by [McDonald \(1981\)](#) and then extended by [ten Berge et al. \(1999\)](#): For a structure matrix, $\mathbf{F}$, with intercorrelations of the factors, Φ , then let $\mathbf{L} = \mathbf{F}\Phi^{1/2}$, and $\mathbf{D} = \mathbf{R}^{1/2}\mathbf{L}(\mathbf{L}'\mathbf{C}^{-1}\mathbf{L})^{-1/2}$, then

$$\mathbf{W} = \mathbf{C}^{-1/2}\mathbf{D}\Phi^{1/2}. \quad (20)$$

Factor score estimates are more precise for more variables/factor

R code

```
set.seed(42)
x8 <- sim.item(nvar=8)
x12 <- sim.item(nvar=12)
x24 <- sim.item(nvar=24)
x36 <- sim.item(nvar=36)
x72 <- sim.item(nvar=72)
f8 <- fa(x8$item, 2)
f12 <- fa(x12$item, 2)
f24 <- fa(x24$item, 2)
f36 <- fa(x36$item, 2)
f72 <- fa(x72$item, 2)
r2.scores.df <- data.frame(f8=f8$R2.scores , f12 =f12$R2.scores,
                           f24=f24$R2.scores, f36=f36$R2.scores, f72=f72$R2.scores)
R2.scores <- round( t(r2.scores.df) , 2)
R2.scores.df <-data.frame(R2.scores, nv=c(8, 12, 24, 36, 72) ,
                          lv =log(c(8, 12, 24, 36, 72)))
```

#Factor score determinancy increases with number of markers/factor

	MR1	MR2	nv	lv
f8	0.70	0.70	8	2.08
f12	0.76	0.76	12	2.48
f24	0.87	0.87	24	3.18
f36	0.92	0.92	36	3.58
f72	0.95	0.95	72	4.28

Show the factor score correlations with the latent variables

R code

```
cor2(f8$scores, x8$theta)
cor2(f12$scores, x12$theta)
cor2(f24$scores, x24$theta)
cor2(f36$scores, x36$theta)
cor2(f72$scores, x72$theta)
```

```
> cor2(f8$scores, x8$theta)
      truex truey
MR1 -0.83  0.00
MR2 -0.01  0.82
> cor2(f12$scores, x12$theta)
      truex truey
MR1  0.89 -0.02
MR2 -0.05  0.89
> cor2(f24$scores, x24$theta)
      truex truey
MR1  0.01  0.93
MR2 -0.92  0.02
> cor2(f36$scores, x36$theta)
      truex truey
MR1  0.96 -0.06
MR2 -0.04  0.95
> cor2(f72$scores, x72$theta)
      truex truey
MR1  0.98 -0.11
MR2 -0.10  0.97
```

```
#Factor score determinancy increases with number of
                                markers/factor
R.scores.df <-data.frame(sqrt(R2.scores), '
                        nv=c(8,12,24,36,72),
                        lv =log(c(8,12,24,36,72)))
round(R.scores.df,2)
      MR1  MR2  nv  lv
f8  0.85  0.84  8  2.08
f12 0.88  0.88 12  2.48
f24 0.93  0.93 24  3.18
f36 0.96  0.95 36  3.58
f72 0.97  0.97 72  4.28
```

Simple Structure

The original solution of a principal components or principal axes factor analysis is a set of vectors that best account for the observed covariance or correlation matrix, and where the components or factors account for progressively less and less variance. But such a solution, although maximally efficient in describing the data, is rarely easy to interpret. But what makes a structure easy to interpret? Thurstone's answer, *simple structure*, consists of five rules (Thurstone, 1947, p 335):

- (1) *Each row of the oblique factor matrix $\mathbf{V}$ should have at least one zero.*
- (2) *For each column p of the factor matrix $\mathbf{V}$ there should be a distinct set of r linearly independent tests whose factor loadings v_{ip} are zero.*
- (3) *For every pair of columns of $\mathbf{V}$ there should be several tests whose entries v_{ip} vanish in one column but not in the other.*
- (4) *For every pair of columns of $\mathbf{V}$, a large proportion of the tests should have zero entries in both columns. This applies to factor problems with four or five or more common factors.*
- (5) *For every pair of columns there should preferably be only a small number of tests with non-vanishing entries in both columns.*

Thurstone proposed to rotate the original solution to achieve simple structure.

Simple structure

A matrix is said to be *rotated* if it is multiplied by a matrix of orthogonal vectors that preserves the communalities of each variable. Just as the original matrix was orthogonal, so is the rotated solution. For two factors, the *rotation* matrix $\mathbf{T}$ will rotate the two factors θ radians in a counterclockwise direction.

$$\mathbf{T} = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix} \quad (21)$$

Generalizing equation 21 to larger matrices is straight forward:

$$\mathbf{T} = \begin{pmatrix} 1 & \dots & 0 & \dots & 0 & \dots & 0 \\ 0 & \dots & \cos(\theta) & \dots & \sin(\theta) & \dots & 0 \\ \dots & \dots & 0 & 1 & 0 & \dots & 0 \\ 0 & \dots & -\sin(\theta) & \dots & \cos(\theta) & \dots & 0 \\ \dots & \dots & 0 & \dots & 0 & \dots & \dots \\ 0 & \dots & 0 & \dots & 0 & \dots & 1 \end{pmatrix}. \quad (22)$$

Rotating to simple structure

When $\mathbf{F}$ is post-multiplied by $\mathbf{T}$, $\mathbf{T}$ will rotate the i^{th} and j^{th} columns of $\mathbf{F}$ by θ radians in a counterclockwise direction.

$$\mathbf{F}_r = \mathbf{F}\mathbf{T} \quad (23)$$

The `factor.rotate` function from the **psych** package will do this rotation for arbitrary angles (in degrees) for any pairs of factors. This is useful if there is a particular rotation that is desired. An entire package devoted to rotations is the *GPArotation* by Robert Jennrich ([Jennrich, 2004](#)).

Analytic Simple Structure

As pointed out by [Carroll \(1953\)](#) when discussing Thurstone's (1947) simple structure as a rotational criterion "it is obvious that there could hardly be any single mathematical expression which could embody all these characteristics." (p 24). Carroll's solution to this was to minimize the sum of the inner products of the squared (rotated) loading matrix. An alternative, discussed by [Ferguson \(1954\)](#) is to consider the *parsimony* of a group of n tests with r factors to be defined as the average parsimony of the individual tests (I_j) where

$$I_j = \sum_m^r a_{jm}^4 \quad (24)$$

(the squared communality) and thus the average parsimony is

$$I_{\cdot} = n^{-1} \sum_j^n \sum_m^r a_{jm}^4$$

and to choose a rotation that maximizes parsimony.

Rotation to parsimony

Parsimony as defined in equation 24 is a function of the variance as well as the mean of the squared loadings of a particular test on all the factors. For fixed communality h^2 , it will be maximized if all but one loading is zero; a variable's parsimony will be maximal if one loading is 1.0 and the rest are zero. In path notation, parsimony is maximized if one and only one arrow is associated with a variable. This criterion, as well as the criterion of maximum variance taken over factors has been operationalized as the *quartimax* criterion by Neuhaus and Wrigley (1954). As pointed out by Kaiser (1958), the criterion can rotate towards a solution with one general factor, ignoring other, smaller factors.

Varimax and alternatives

If a general factor is not desired, an alternative measure of the parsimony of a factor, similar to equation 24 is to maximize the variance of the squared loadings taken over items instead of over factors. This, the *varimax* criterion was developed by Kaiser (1958) to avoid the tendency to yield a general factor. Both of these standard rotations as well as many others are available in the **GPArotation** package of rotations and transformations which uses the *Gradient Projection Algorithms* developed by Jennrich (2001, 2002, 2004).

Harman 8 physical measures

```
> data(Harman23.cor)
> lower.mat(Harman23.cor$cov)
```

	height	arm.s	form	lwr.l	weght	btr.d	chst.g	chst.w
height	1.00							
arm.span	0.85	1.00						
forearm	0.80	0.88	1.00					
lower.leg	0.86	0.83	0.80	1.00				
weight	0.47	0.38	0.38	0.44	1.00			
bitro.diameter	0.40	0.33	0.32	0.33	0.76	1.00		
chest.girth	0.30	0.28	0.24	0.33	0.73	0.58	1.00	
chest.width	0.38	0.42	0.34	0.36	0.63	0.58	0.54	1.00

Two solutions – loadings change, goodness of fits do not

```
> f2 <- fa(Harman23.cor$cov, 2, rotate="none")
> f2
```

```
Factor Analysis using method = minres
Call: fa(r = Harman23.cor$cov, nfactors = 2,
  rotate = "none")
```

Standardized loadings (pattern **matrix**)

	MR1	MR2	h2	u2
height	0.89	-0.19	0.83	0.17
arm.span	0.89	-0.31	0.89	0.11
forearm	0.86	-0.30	0.83	0.17
lower.leg	0.87	-0.22	0.80	0.20
weight	0.67	0.67	0.89	0.11
bitro.diameter	0.56	0.58	0.65	0.35
chest.girth	0.50	0.59	0.59	0.41
chest.width	0.56	0.40	0.47	0.53

	MR1	MR2
SS loadings	4.40	1.56
Proportion Var	0.55	0.19
Cumulative Var	0.55	0.74

Test of the hypothesis that 2 factors are sufficient.

The root **mean** square of the **residuals** (RMSR) is 0.02

The **df** corrected root **mean** square of the **residuals** is 0.03

Fit based upon **off** diagonal values = 1

```
> f2 <- fa(Harman23.cor$cov, 2, rotate="varimax")
> f2
```

```
Factor Analysis using method = minres
Call: fa(r = Harman23.cor$cov, nfactors = 2,
  rotate = "varimax")
```

Standardized loadings (pattern **matrix**)

	MR1	MR2	h2	u2
height	0.86	0.30	0.83	0.17
arm.span	0.92	0.20	0.89	0.11
forearm	0.89	0.19	0.83	0.17
lower.leg	0.86	0.26	0.80	0.20
weight	0.22	0.92	0.89	0.11
bitro.diameter	0.18	0.78	0.65	0.35
chest.girth	0.12	0.76	0.59	0.41
chest.width	0.27	0.63	0.47	0.53

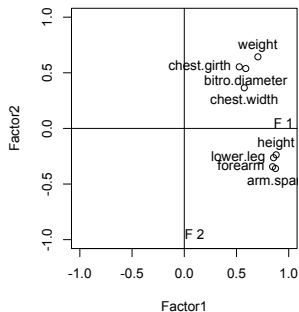
	MR1	MR2
SS loadings	3.30	2.66
Proportion Var	0.41	0.33
Cumulative Var	0.41	0.74

The root **mean** square of the **residuals** (RMSR) is 0.02

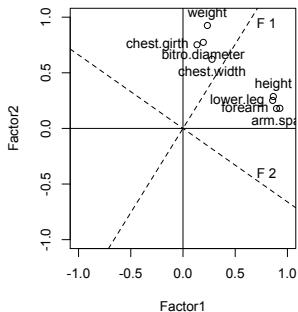
The **df** corrected root **mean** square of the **residuals** is 0.03

Alternative rotations

Unrotated



Varimax rotated



Oblique transformations

Many of those who use factor analysis use it to identify theoretically meaningful constructs which they have no reason to believe are orthogonal. This has led to the use of *oblique transformations* which allow the factors to be correlated. Although the term rotation is sometimes used for both *orthogonal* and *oblique* solutions, in the oblique case the factor matrix is not rotated so much as *transformed*.

Oblique transformations lead to the distinction between the *factor pattern* and *factor structure* matrices. The *factor pattern* matrix is the set of *regression weights* (loadings) from the latent factors to the observed variables. The *factor structure* matrix is the matrix of *correlations* between the factors and the observed variables. If the factors are uncorrelated, structure and pattern are identical. But, if the factors are correlated, the structure matrix (**S**) is the pattern matrix (**F**) times the factor intercorrelations ϕ

$$\mathbf{S} = \mathbf{F}\phi \iff \mathbf{F} = \mathbf{S}\phi^{-1}:$$

An oblique transformation of the Harman 8 physical variables

```
> f2t <- fa(Harman23.cor$cov, 2, rotate = "oblimin", n.obs = 305)
> print(f2t)
```

Factor Analysis using method = minres

Call: fa(r = Harman23.cor\$cov, nfactors = 2, rotate = "oblimin", n.obs = 305)

	item	MR1	MR2	h2	u2
height	1	0.87	0.08	0.84	0.16
arm.span	2	0.96	-0.05	0.89	0.11
forearm	3	0.93	-0.04	0.83	0.17
lower.leg	4	0.88	0.04	0.81	0.19
weight	5	0.01	0.94	0.89	0.11
bitro.diameter	6	0.00	0.80	0.64	0.36
chest.girth	7	-0.06	0.79	0.59	0.41
chest.width	8	0.13	0.62	0.47	0.53

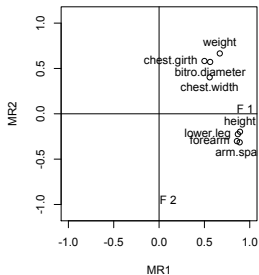
	MR1	MR2
SS loadings	3.37	2.58
Proportion Var	0.42	0.32
Cumulative Var	0.42	0.74

With **factor** correlations of

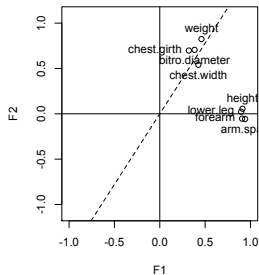
	MR1	MR2
MR1	1.00	0.46
MR2	0.46	1.00

Oblique Transformations

Unrotated

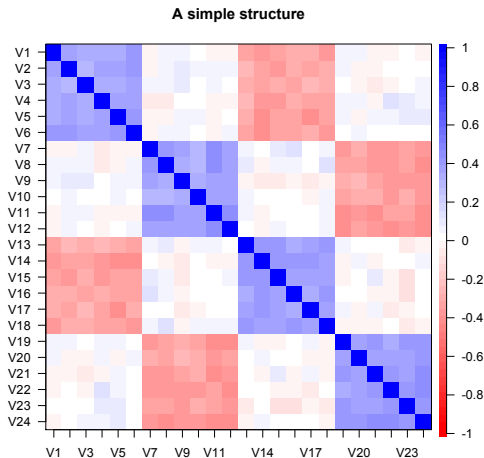


Oblique Transformation



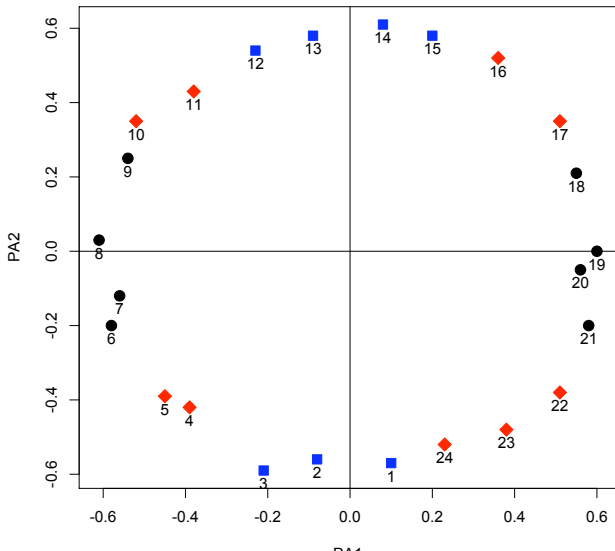
Another way to show simple structure

```
> simp24 <- sim.item(24, circum=FALSE)
> cor.plot(cor(simp24), main="A_simple_structure")
```



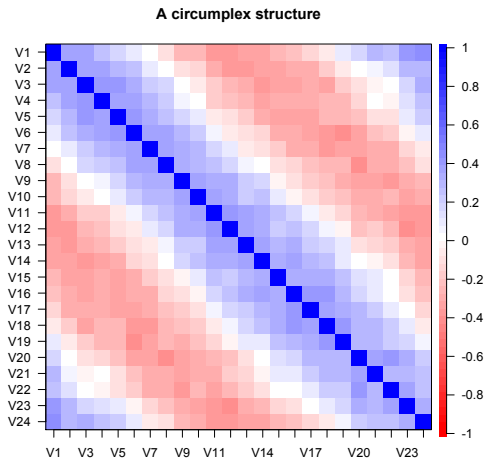
A circumplex is one alternative to simple structure

Circumplex structure



Another way of showing a circumplex – cor.plot

```
> circ24 <- sim.item(24, circum=TRUE)
> cor.plot(cor(circ24), main="A_circumplex_structure")
```



The Thurstone 9 variable problem

```
> lower.mat(Thurstone)
```

	Sntnc	Vcblr	Snt.C	Frs.L	4.L.W	Sffxs	Ltt.S	Pdgrs	Ltt.G
Sentences	1.00								
Vocabulary	0.83	1.00							
Sent. Completion	0.78	0.78	1.00						
First. Letters	0.44	0.49	0.46	1.00					
4. Letter. Words	0.43	0.46	0.42	0.67	1.00				
Suffixes	0.45	0.49	0.44	0.59	0.54	1.00			
Letter. Series	0.45	0.43	0.40	0.38	0.40	0.29	1.00		
Pedigrees	0.54	0.54	0.53	0.35	0.37	0.32	0.56	1.00	
Letter. Group	0.38	0.36	0.36	0.42	0.45	0.32	0.60	0.45	1.00

Three factors from Thurstone 9 variables

```
> f3 <- fa(Thurstone,3)
> f3
```

Factor Analysis using method = minres

Call: fa(r = Thurstone, nfactors = 3)

Standardized loadings (pattern **matrix**) based upon correlation **matrix**

	MR1	MR2	MR3	h2	u2
Sentences	0.91	-0.04	0.04	0.82	0.18
Vocabulary	0.89	0.06	-0.03	0.84	0.16
Sent. Completion	0.83	0.04	0.00	0.73	0.27
First. Letters	0.00	0.86	0.00	0.73	0.27
4. Letter. Words	-0.01	0.74	0.10	0.63	0.37
Suffixes	0.18	0.63	-0.08	0.50	0.50
Letter. Series	0.03	-0.01	0.84	0.72	0.28
Pedigrees	0.37	-0.05	0.47	0.50	0.50
Letter. Group	-0.06	0.21	0.64	0.53	0.47

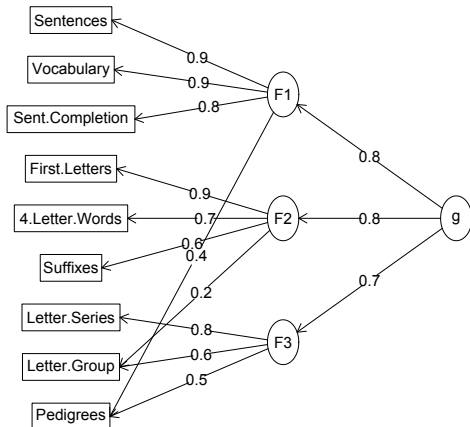
	MR1	MR2	MR3
SS loadings	2.64	1.86	1.50
Proportion Var	0.29	0.21	0.17
Cumulative Var	0.29	0.50	0.67

With **factor** correlations of

	MR1	MR2	MR3
MR1	1.00	0.59	0.54
MR2	0.59	1.00	0.52
MR3	0.54	0.52	1.00

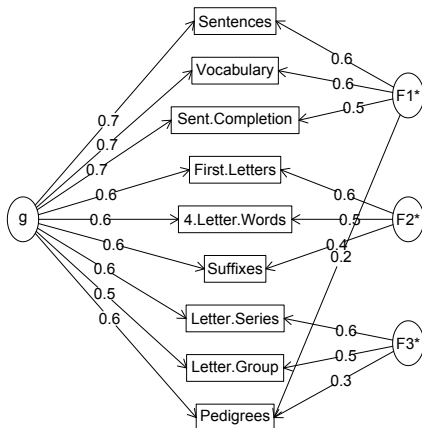
A hierarchical/multilevel solution to the Thurstone 9 variables

Hierarchical (multilevel) Structure



A bifactor solution using the Schmid Leiman transformation

Omega with Schmid Leiman Transformation



Find the Schmid-Leiman solution

```
fload <- f3$loadings
g <- fa(f3$Phi)$loadings
colnames(g) <- 'g'
gload <- fload %*% g
lg <- diag(drop(gload))
Fstar <- t(t(fload) %*% lg)
colnames(Fstar) <- paste("F", 1:3, "*", sep = "")
schmid <- cbind(gload, Fstar)
```

```
round(schmid, 2)
```

	g	F1*	F2*	F3*
Sentences	0.71	0.64	-0.03	0.03
Vocabulary	0.73	0.64	0.05	-0.02
Sent. Completion	0.68	0.57	0.03	0.00
First. Letters	0.65	0.00	0.55	0.00
4. Letter. Words	0.62	-0.01	0.46	0.06
Suffixes	0.56	0.10	0.35	-0.05
Letter. Series	0.59	0.02	-0.01	0.49
Pedigrees	0.58	0.22	-0.03	0.27
Letter. Group	0.54	-0.03	0.11	0.34

```
>
```

1. Find the loadings from the fa
2. Factor the interfactor correlations
3. Multiply these two together
4. create a diagonal matrix
5. Find the residual loadings
6. Combine everything

Schmid Leiman $\neq$ bifactor solution

```
om <- omega(Thurstone)
```

Schmid Leiman Factor loadings greater than 0.2

	h2	u2	p2	g	F1*	F2*	F3*
Sentences	0.82	0.18	0.61	0.71	0.57		
Vocabulary	0.84	0.16	0.63	0.73	0.55		
Sent.Completion	0.73	0.27	0.63	0.68	0.52		
First.Letters	0.73	0.27	0.57	0.65		0.56	
4.Letter.Words	0.63	0.37	0.61	0.62		0.49	
Suffixes	0.50	0.50	0.63	0.56		0.41	
Letter.Series	0.61	0.72	0.28	0.59			
Pedigrees	0.34	0.50	0.50	0.58	0.23		
Letter.Group	0.46	0.53	0.47	0.54			

With eigenvalues of:

g	F1*	F2*	F3*
3.58	0.96	0.74	0.71

```
f4 <- fa(Thurstone,4,rotate="bifactor")
```

Factor Analysis using method = minres

Call: fa(r = Thurstone, nfactors = 4, rotate = "bifactor")
Standardized loadings (pattern **matrix**) based upon initial factor scores

	MR1	MR2	MR3	MR4
Sentences	0.91	-0.25		
Vocabulary	0.88	-0.16	-0.07	
Sent.Completion	0.83	-0.16	-0.05	
First.Letters	0.63	0.50	-0.30	
4.Letter.Words	0.60			
Suffixes	0.58	0.29	-0.28	
Letter.Series	0.58	0.34		
Pedigrees	0.64	0.09	0.30	
Letter.Group	0.52	0.40		

	MR1	MR2	MR3	MR4
SS loadings	4.40	0.95	0.67	0.08
Proportion Var	0.49	0.11	0.07	0.01
Cumulative Var	0.49	0.59	0.67	0.68
Proportion Explained	0.72	0.16	0.11	0.01
Cumulative Proportion	0.72	0.88	0.99	1.00

How many factors – no right answer, one wrong answer

1. Statistical

- Extracting factors until the χ^2 of the residual matrix is not significant.
- Extracting factors until the change in χ^2 from factor n to factor $n+1$ is not significant.

2. Rules of Thumb

- Parallel Extracting factors until the eigenvalues of the real data are less than the corresponding eigenvalues of a random data set of the same size (*parallel analysis*)
- Plotting the magnitude of the successive eigenvalues and applying the *scree test*.

3. Interpretability

- Extracting factors as long as they are interpretable.
- Using the *Very Simple Structure* Criterion (VSS)
- Using the Minimum Average Partial criterion (MAP).

4. Eigen Value of 1 rule

Simulate 2 factor data

Using the sim.item function

```
> set.seed(42) #to generate a reproducible example
> my.data <- sim.item(12)
> my.cor <- cor(my.data)
> round(my.cor, 2)
```

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	
V12												
V1	1.00	0.36	0.38	-0.01	0.05	0.03	-0.35	-0.40	-0.41	0.06	0.02	
0.01												
V2	0.36	1.00	0.37	-0.04	-0.02	0.01	-0.37	-0.34	-0.36	0.07	0.03	
0.01												
V3	0.38	0.37	1.00	-0.01	0.01	0.01	-0.38	-0.39	-0.32	0.01	0.05	-0.1
V4	-0.01	-0.04	-0.01	1.00	0.34	0.37	-0.09	0.00	0.00	-0.33	-0.37	-0.3
V5	0.05	-0.02	0.01	0.34	1.00	0.35	-0.01	0.08	0.02	-0.32	-0.35	-0.3
V6	0.03	0.01	0.01	0.37	0.35	1.00	-0.05	0.11	-0.03	-0.39	-0.32	-0.3
V7	-0.35	-0.37	-0.38	-0.09	-0.01	-0.05	1.00	0.34	0.32	-0.04	0.02	
0.08												
V8	-0.40	-0.34	-0.39	0.00	0.08	0.11	0.34	1.00	0.39	-0.11	-0.12	-0.0
V9	-0.41	-0.36	-0.32	0.00	0.02	-0.03	0.32	0.39	1.00	-0.06	-0.01	
0.00												
V10	0.06	0.07	0.01	-0.33	-0.32	-0.39	-0.04	-0.11	-0.06	1.00	0.41	
0.36												
V11	0.02	0.03	0.05	-0.37	-0.35	-0.32	0.02	-0.12	-0.01	0.41	1.00	
0.39												

Multiple ways to determine how many factors are in the data

No one answer. Many are good, one should be avoided.

1. Statistical tests

- χ^2 test of residuals (sensitive to sample size and non-normality of data)
- χ^2 test of change from $nf=n$ to $nf=n+1$ (sensitive to sample size)
- RMSEA, BIC, AIC, SABIC are not as sensitive to sample size, but are to non-normality

2. Rules of Thumb

- Scree Test of eigen values ([Cattell, 1966](#))
- Minimum Average Partial (MAP) ([Velicer, 1976](#))
- Very Simple Structure ([Revelle and Rocklin, 1979](#))
- Parallel Analysis of random data ([Horn, 1965](#))
- As many as can be interpreted

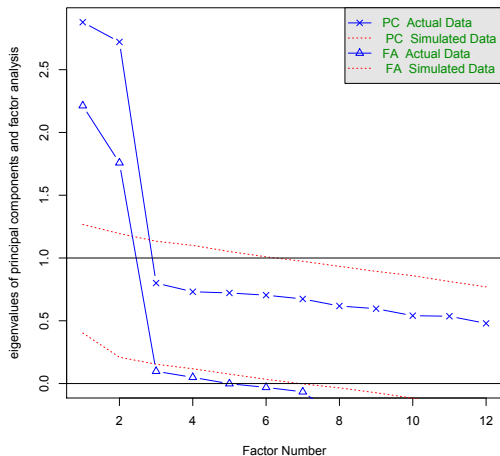
3. One test to avoid: Eigen value of 1 (Many programs default to this)

How many factors in my.cor

```
> fa.parallel(my.cor,n.obs=500)
```

Parallel analysis suggests that the number of factors = 2
and the number of components = 2

Parallel Analysis Scree Plots



Try Very Simple Structure as well as MAP

```
> vss(my.cor, n.obs=500)
```

Very Simple Structure

Call: VSS(x = x, n = n, rotate = rotate, diagonal = diagonal, fm = fm,
n.obs = n.obs, **plot = plot, title = title**)

VSS complexity 1 achieves a maximum of 0.74 with 3 factors

VSS complexity 2 achieves a maximum of 0.8 with 8 factors

The Velicer MAP criterion achieves a minimum of 0.02 with 2 factors

Velicer MAP

```
[1] 0.05 0.02 0.03 0.05 0.07 0.10 0.13 0.19
```

Very Simple Structure Complexity 1

```
[1] 0.39 0.74 0.74 0.63 0.70 0.66 0.58 0.57
```

Very Simple Structure Complexity 2

```
[1] 0.00 0.75 0.76 0.78 0.79 0.79 0.80 0.80
```

Even more VSS output

```
> my.vss <- vss(my.cor, n=12)
> my.vss
```

n.obs was not specified and was arbitrarily **set** to 1000. This only affects the chi square values.
Very Simple Structure

Call: vss(x = my.cor, n = 12)

VSS complexity 1 achieves a maximum of 0.74 with 3 factors

VSS complexity 2 achieves a maximum of 0.81 with 10 factors

The Velicer MAP achieves a minimum of NA with 2 factors

BIC achieves a minimum of NA with 2 factors

Sample Size adjusted BIC achieves a minimum of NA with 2 factors

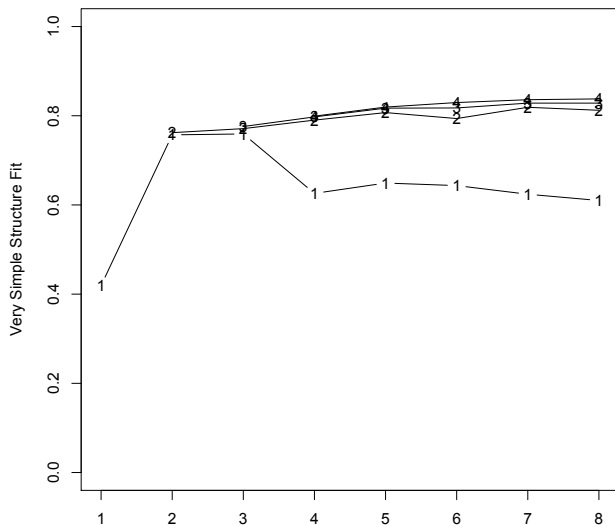
Statistics **by** number of factors

	vss1	vss2	map	dof	chisq	prob	sqresid	fit	RMSEA	BIC	SABIC	complex
1	0.39	0.00	0.055	54	1.3e+03	1.5e-229	12.2	0.39	0.150	894	1065.3	1.0
2	0.74	0.75	0.021	43	1.1e+02	9.0e-08	5.0	0.75	0.040	-187	-50.6	1.0
3	0.74	0.76	0.033	33	7.6e+01	3.1e-05	4.7	0.76	0.036	-152	-47.2	1.1
4	0.63	0.78	0.049	24	5.3e+01	6.0e-04	4.2	0.79	0.035	-113	-36.7	1.2
5	0.70	0.79	0.069	16	3.4e+01	5.6e-03	3.9	0.80	0.034	-77	-25.8	1.3
6	0.66	0.79	0.096	9	1.1e+01	2.6e-01	3.5	0.82	0.016	-51	-22.3	1.3
7	0.58	0.80	0.130	3	1.8e+00	6.2e-01	3.3	0.83	0.000	-19	-9.4	1.5
8	0.57	0.80	0.186	-2	8.4e-02	NA	2.8	0.86	NA	NA	NA	1.5
9	0.43	0.74	0.278	-6	8.8e-07	NA	2.6	0.87	NA	NA	NA	1.9
10	0.62	0.81	0.456	-9	1.7e-09	NA	3.1	0.85	NA	NA	NA	1.8
11	0.55	0.81	1.000	-11	0.0e+00	NA	3.0	0.85	NA	NA	NA	1.8
12	0.55	0.81	NA	-12	0.0e+00	NA	3.0	0.85	NA	NA	NA	1.8

```
>
```

Examine the output

Very Simple Structure



nfactors does multiple tests

```
> nfactors(my.cor)
```

n.obs was not specified and was arbitrarily **set** to 1000. This only affects the chi square values.

Number of factors

Call: vss(x = x, n = n, rotate = rotate, diagonal = diagonal, fm = fm,
n.obs = n.obs, **plot** = FALSE, **title** = title, use = use, **cor** = cor)

VSS complexity 1 achieves a maximum of 0.74 with 3 factors

VSS complexity 2 achieves a maximum of 0.81 with 10 factors

The Velicer MAP achieves a minimum of 0.02 with 2 factors

Empirical BIC achieves a minimum of -209.08 with 2 factors

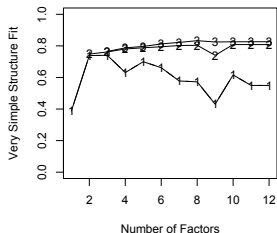
Sample Size adjusted BIC achieves a minimum of -50.56 with 2 factors

Statistics **by** number of factors

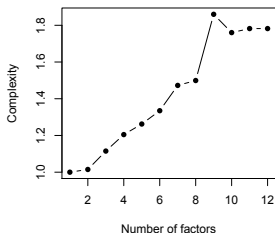
	vss1	vss2	map	dof	chisq	prob	sqresid	fit	RMSEA	BIC	SABIC	complex	eChisq
SPMR eCRMS eBIC													
1	0.39	0.00	0.055	54	1.3e+03	1.5e-229	12.2	0.39	0.150	894	1065.3	1.0	3.8e+03
2	0.74	0.75	0.021	43	1.1e+02	9.0e-08	5.0	0.75	0.040	-187	-50.6	1.0	8.8e+01
3	0.74	0.76	0.033	33	7.6e+01	3.1e-05	4.7	0.76	0.036	-152	-47.2	1.1	5.7e+01
4	0.63	0.78	0.049	24	5.3e+01	6.0e-04	4.2	0.79	0.035	-113	-36.7	1.2	4.2e+01
5	0.70	0.79	0.069	16	3.4e+01	5.6e-03	3.9	0.80	0.034	-77	-25.8	1.3	2.4e+01
-86													
6	0.66	0.79	0.096	9	1.1e+01	2.6e-01	3.5	0.82	0.016	-51	-22.3	1.3	8.4e+00
-54													
7	0.58	0.80	0.130	3	1.8e+00	6.2e-01	3.3	0.83	0.000	-19	-9.4	1.5	1.4e+00
-19													
8	0.57	0.80	0.186	-2	8.4e-02	NA	2.8	0.86	NA	NA	NA	1.5	6.8e-02
NA NA													
9	0.43	0.74	0.278	-6	8.8e-07	NA	2.6	0.87	NA	NA	NA	1.9	6.4e-07
NA NA													
10	0.62	0.81	0.456	-9	1.7e-09	NA	3.1	0.85	NA	NA	NA	1.8	9.4e-10
NA NA													

nfactors shows a number of fit statistics

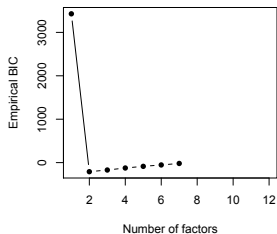
Very Simple Structure



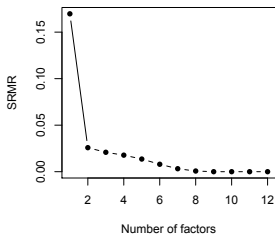
Complexity



Empirical BIC



Root Mean Residual



Extract 2 factors –part 1

```
> fa(my.cor,2,n.obs=500)
```

Factor Analysis using method = minres

Call: fa(r = my.cor, nfactors = 2, n.obs = 500)

Standardized loadings based upon correlation **matrix**

	MR1	MR2	h2	u2
V1	0.64	-0.02	0.41	0.59
V2	0.59	0.02	0.35	0.65
V3	0.61	-0.04	0.37	0.63
V4	0.03	-0.58	0.34	0.66
V5	0.01	-0.55	0.30	0.70
V6	0.03	-0.60	0.36	0.64
V7	-0.58	0.08	0.34	0.66
V8	-0.62	-0.10	0.40	0.60
V9	-0.59	0.00	0.35	0.65
V10	0.07	0.61	0.39	0.61
V11	0.03	0.63	0.39	0.61
V12	-0.06	0.57	0.33	0.67

	MR1	MR2
SS loadings	2.21	2.12
Proportion Var	0.18	0.18
Cumulative Var	0.18	0.36

2 artificial factors part 2

With **factor** correlations of

MR1 MR2

MR1 1.00 0.04

MR2 0.04 1.00

Test of the hypothesis that 2 factors are sufficient.

The degrees of freedom **for** the **null model** are 66 and the objective **function** is 2.52 with Chi Square of 1246.71

The degrees of freedom **for** the **model** are 43 and the objective **function** value is 0.11

The root **mean** square of the **residuals** is 0.02

The **df** corrected root **mean** square of the **residuals** is 0.03

The number of observations was 500 with Chi Square = 54.56 with prob < 0.11

Tucker Lewis Index of factoring reliability = 0.985

RMSEA **index** = 0.024 and the 90 % confidence intervals are 0.023 0.026

BIC = -212.67

Fit based upon **off** diagonal values = 0.99

Measures of **factor** score adequacy

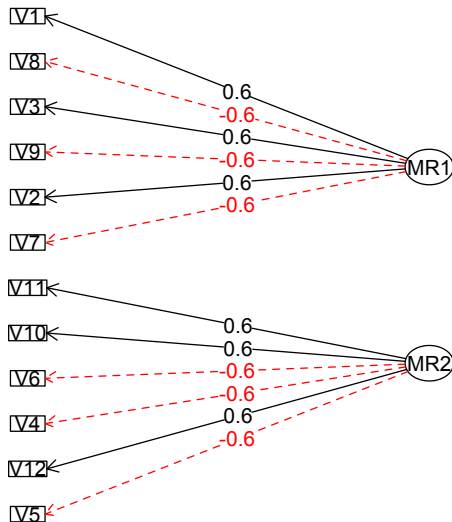
MR1 MR2

Correlation of scores with factors

0.88 0.88

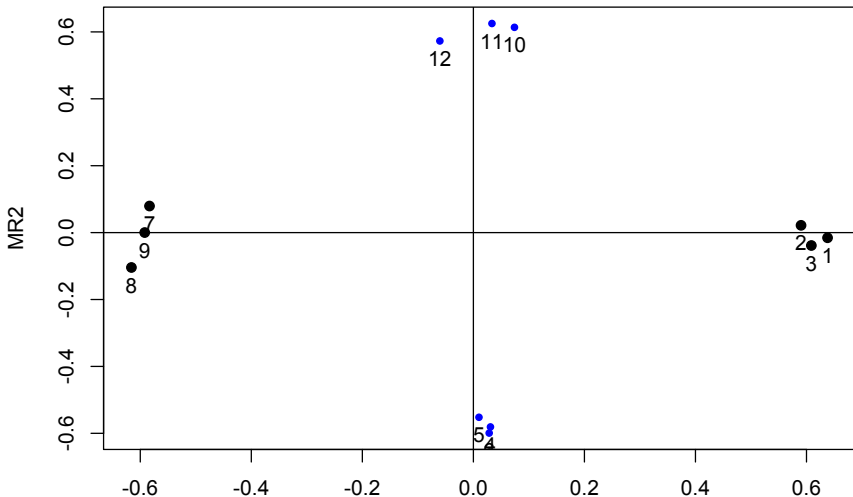
The factor diagram shows the structure

Factor Analysis



The factor plot also shows the structure

Factor Analysis



An alternative data structure is a circumplex

Seen in measures of emotion, interpersonal problems

```
> circ <- sim.circ(12)
> f2 <- fa(circ, 2)
> fa.plot(f2, title="A_circumplex_structure")
> fa.diagram(f2, simple=FALSE, main="A_circumplex_structure")
```

Factor Analysis using method =
minres

Call: fa(r = circ, nfactors = 2)

Standardized loadings (pattern **matrix**)

	MR1	MR2	h2	u2	com
1	0.27	-0.48	0.30	0.70	1.6
2	-0.07	-0.61	0.38	0.62	1.0
3	-0.40	-0.39	0.32	0.68	2.0
4	-0.54	-0.26	0.36	0.64	1.4
5	-0.67	0.10	0.46	0.54	1.0
6	-0.44	0.36	0.32	0.68	1.9
7	-0.17	0.54	0.32	0.68	1.2
8	0.04	0.59	0.35	0.65	1.0
9	0.35	0.46	0.34	0.66	1.9
10	0.60	0.22	0.41	0.59	1.3
11	0.60	-0.05	0.37	0.63	1.0
12	0.53	-0.33	0.38	0.62	1.7

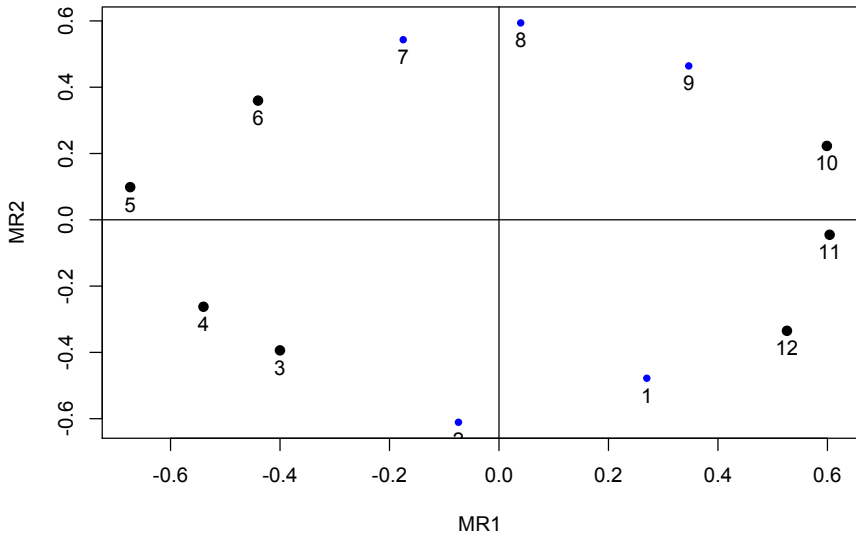
	MR1	MR2
SS loadings	2.33	1.99
Proportion Var	0.19	0.17
Cumulative Var	0.19	0.36
Proportion Explained	0.54	0.46
Cumulative Proportion	0.54	1.00

With **factor** correlations of

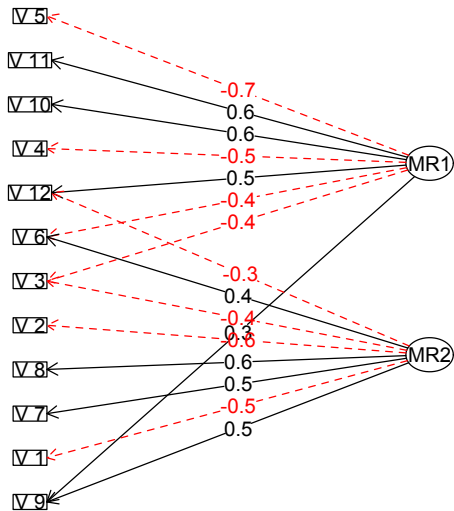
	MR1	MR2
MR1	1.00	0.01
MR2	0.01	1.00

Mean item complexity = 1.4

A circumplex structure



A circumplex structure



9 cognitive tests from Thurstone

```
> lowerMat(Thurstone)
```

	Sntnc	Vcblr	Snt.C	Frs.L	4.L.W	Sffxs	Ltt.S	Pdgrs	Ltt.G
Sentences	1.00								
Vocabulary	0.83	1.00							
Sent. Completion	0.78	0.78	1.00						
First. Letters	0.44	0.49	0.46	1.00					
4. Letter. Words	0.43	0.46	0.42	0.67	1.00				
Suffixes	0.45	0.49	0.44	0.59	0.54	1.00			
Letter. Series	0.45	0.43	0.40	0.38	0.40	0.29	1.00		
Pedigrees	0.54	0.54	0.53	0.35	0.37	0.32	0.56	1.00	
Letter. Group	0.38	0.36	0.36	0.42	0.45	0.32	0.60	0.45	1.00

Choices in dimension reduction

1. Factors or Components

- Components are maximally efficient to describe the data (including the error)
- Factors model the shared common variance, but not the errors

2. How many factors to extract

3. Which factor extraction technique

- maximum likelihood is "optimal" but not if model of mean residual = 0 is false
- minres minimizes the residuals using Ordinary Least Squares – fits are almost as good as mle
- principal axis is an iterative procedure used by SPSS
- minchi minimizes the sample size weighted residual if number of pairwise observations differ

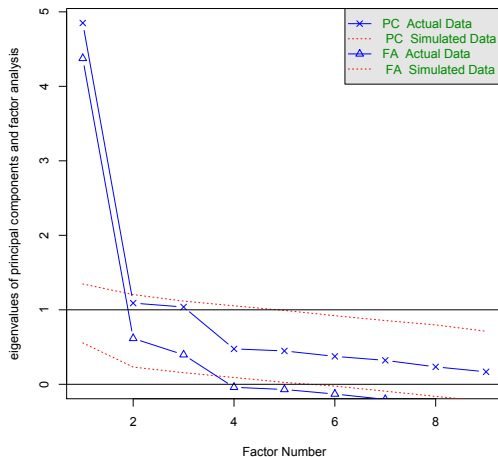
4. rotation (orthogonal) or transformation (oblique)

- Orthogonal rotations (e.g.,) varimax, quartimax, bifactor
- Oblique transformations (e.g.,) oblimin, oblimax, geomin, biquaritimn,
- Higher order structures (e.g.,) schmid leiman, bifactor

9 mental tests from Thurstone

```
data(bifactor)
fa.parallel(Thurstone, n.obs=213)
```

Parallel Analysis Scree Plots



Extract 3 factors

```
> fa3 <- fa(Thurstone,3,n.obs=213)
```

```
> fa3
```

Factor Analysis using method = minres

Call: fa(r = Thurstone, nfactors = 3, n.obs = 213)

Standardized loadings based upon correlation **matrix**

	MR1	MR2	MR3	h2	u2
Sentences	0.91	-0.04	0.04	0.82	0.18
Vocabulary	0.89	0.06	-0.03	0.84	0.16
Sent.Completion	0.83	0.04	0.00	0.73	0.27
First.Letters	0.00	0.86	0.00	0.73	0.27
4.Letter.Words	-0.01	0.74	0.10	0.63	0.37
Suffixes	0.18	0.63	-0.08	0.50	0.50
Letter.Series	0.03	-0.01	0.84	0.72	0.28
Pedigrees	0.37	-0.05	0.47	0.50	0.50
Letter.Group	-0.06	0.21	0.64	0.53	0.47

	MR1	MR2	MR3
SS loadings	2.64	1.86	1.50
Proportion Var	0.29	0.21	0.17
Cumulative Var	0.29	0.50	0.67

Thurstone 3 factors part 2

With **factor** correlations of

	MR1	MR2	MR3
MR1	1.00	0.59	0.54
MR2	0.59	1.00	0.52
MR3	0.54	0.52	1.00

Test of the hypothesis that 3 factors are sufficient.

The degrees of freedom **for** the **null model** are 36 and the objective **function** is 5.2 with Chi Square of 1081.97

The degrees of freedom **for** the **model** are 12 and the objective **function** was 0.01

The root **mean** square of the **residuals** is 0

The **df** corrected root **mean** square of the **residuals** is 0.01

The number of observations was 213 with Chi Square = 2.82 with prob < 1

Tucker Lewis Index of factoring reliability = 1.027

RMSEA **index** = 0 and the 90 % confidence intervals are 0 0.023

BIC = -61.51

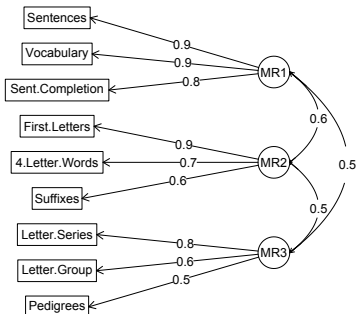
Fit based upon **off** diagonal values = 1

Measures of **factor** score adequacy

A factor diagram

```
fa3 <- fa(Thurstone,3,n.obs=213)
```

Factor Analysis



Thurstone, 3 factors Varimax rotated

```
> v3 <- fa(Thurstone,3,rotate="Varimax",n.obs=213)
> fa.diagram(v3)
> v3
Factor Analysis using method = minres
Call: fa(r = Thurstone, nfactors = 3, n.obs = 213, rotate = "Varimax")
Standardized loadings based upon correlation matrix
```

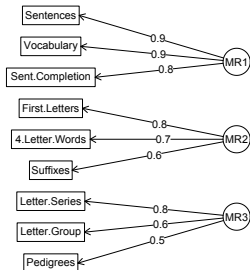
	MR1	MR2	MR3	h2	u2
Sentences	0.86	0.20	0.22	0.82	0.18
Vocabulary	0.85	0.27	0.18	0.84	0.16
Sent.Completion	0.80	0.24	0.19	0.73	0.27
First.Letters	0.29	0.78	0.20	0.73	0.27
4.Letter.Words	0.27	0.70	0.26	0.63	0.37
Suffixes	0.36	0.60	0.10	0.50	0.50
Letter.Series	0.28	0.18	0.78	0.72	0.28
Pedigrees	0.48	0.15	0.50	0.50	0.50
Letter.Group	0.20	0.32	0.62	0.53	0.47

	MR1	MR2	MR3
SS loadings	2.73	1.78	1.48
Proportion Var	0.30	0.20	0.16
Cumulative Var	0.30	0.50	0.67

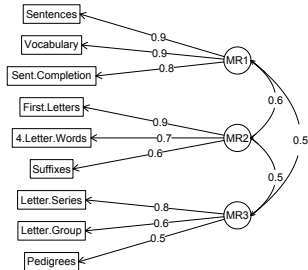
Compare the two solutions

```
> v3 <- fa(Thurstone,3,rotate="Varimax",n.obs=213)
> fa.diagram(v3)
```

Factor Analysis



Factor Analysis



Factor Congruence is a measure of how much the factor loadings agree

- Developed by **Burt (1948)** but known as the “Tucker coefficient” it is just $\frac{\Sigma(F_1 F_2)}{\sqrt{\Sigma F_1^2 \Sigma F_2^2}}$ or the cosine of the angles from zero of the two factors. (Note that this differs from a correlation which centers the vectors on the mean of the vectors.)

> **factor.congruence**(**list**(v3, f3))

	MR1	MR2	MR3	MR1	MR2	MR3
MR1	1.00	0.64	0.63	0.95	0.30	0.28
MR2	0.64	1.00	0.62	0.41	0.92	0.28
MR3	0.63	0.62	1.00	0.41	0.36	0.90
MR1	0.95	0.41	0.41	1.00	0.06	0.09
MR2	0.30	0.92	0.36	0.06	1.00	0.08
MR3	0.28	0.28	0.90	0.09	0.08	1.00

Principal Components of the Thurstone data set

```
> p3 <- principal(Thurstone,3)
> p3
```

Principal Components Analysis

Call: principal(r = Thurstone, nfactors = 3)

Standardized loadings (pattern **matrix**) based upon correlation **matrix**

	RC1	RC2	RC3	h2	u2
Sentences	0.86	0.24	0.23	0.86	0.14
Vocabulary	0.85	0.31	0.19	0.86	0.14
Sent. Completion	0.85	0.26	0.19	0.83	0.17
First. Letters	0.23	0.82	0.23	0.78	0.22
4. Letter. Words	0.18	0.79	0.30	0.75	0.25
Suffixes	0.31	0.77	0.06	0.70	0.30
Letter. Series	0.25	0.16	0.83	0.78	0.22
Pedigrees	0.53	0.08	0.61	0.67	0.33
Letter. Group	0.10	0.31	0.80	0.75	0.25

	RC1	RC2	RC3
SS loadings	2.73	2.25	1.99
Proportion Var	0.30	0.25	0.22
Cumulative Var	0.30	0.55	0.78
Proportion Explained	0.39	0.32	0.29
Cumulative Proportion	0.39	0.71	1.00

Test of the hypothesis that 3 components are sufficient.

The degrees of freedom **for** the **null model** are 36 and the objective **function** was 5.2
 The degrees of freedom **for** the **model** are 12 and the objective **function** was 0.62

Fit based upon **off** diagonal values = 0.98

R has many built in data sets

1. `data(bfi)`
 - 25 personality items from the Big 5
 - Collected as part of the SAPA project
2. Thought to represent 5 dimensions
 - Agreeableness
 - Extraversion
 - Conscientiousness
 - Openness
 - Neuroticism
3. `spi` (SAPA personality inventory) ([Condon, 2018](#))
 - Has Big 5 as well as 'little 27' factor structure
 - The 27 are not facets of the Big 5 but rather a different structural solution

Describe the Big 5

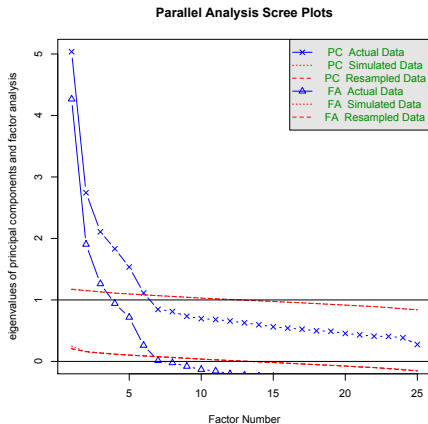
```
> data(bfi)
> describe(bfi)
```

	var	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
A1	1	2784	2.41	1.41	2	2.23	1.48	1	6	5	0.83	-0.31	0.03
A2	2	2773	4.80	1.17	5	4.98	1.48	1	6	5	-1.12	1.05	0.02
A3	3	2774	4.60	1.30	5	4.79	1.48	1	6	5	-1.00	0.44	0.02
A4	4	2781	4.70	1.48	5	4.93	1.48	1	6	5	-1.03	0.04	0.03
A5	5	2784	4.56	1.26	5	4.71	1.48	1	6	5	-0.85	0.16	0.02
C1	6	2779	4.50	1.24	5	4.64	1.48	1	6	5	-0.85	0.30	0.02
C2	7	2776	4.37	1.32	5	4.50	1.48	1	6	5	-0.74	-0.14	0.03
C3	8	2780	4.30	1.29	5	4.42	1.48	1	6	5	-0.69	-0.13	0.02
C4	9	2774	2.55	1.38	2	2.41	1.48	1	6	5	0.60	-0.62	0.03
C5	10	2784	3.30	1.63	3	3.25	1.48	1	6	5	0.07	-1.22	0.03
E1	11	2777	2.97	1.63	3	2.86	1.48	1	6	5	0.37	-1.09	0.03
E2	12	2784	3.14	1.61	3	3.06	1.48	1	6	5	0.22	-1.15	0.03
E3	13	2775	4.00	1.35	4	4.07	1.48	1	6	5	-0.47	-0.47	0.03
E4	14	2791	4.42	1.46	5	4.59	1.48	1	6	5	-0.82	-0.30	0.03
E5	15	2779	4.42	1.33	5	4.56	1.48	1	6	5	-0.78	-0.09	0.03
N1	16	2778	2.93	1.57	3	2.82	1.48	1	6	5	0.37	-1.01	0.03
N2	17	2779	3.51	1.53	4	3.51	1.48	1	6	5	-0.08	-1.05	0.03
N3	18	2789	3.22	1.60	3	3.16	1.48	1	6	5	0.15	-1.18	0.03
N4	19	2764	3.19	1.57	3	3.12	1.48	1	6	5	0.20	-1.09	0.03
N5	20	2771	2.97	1.62	3	2.85	1.48	1	6	5	0.37	-1.06	0.03
O1	21	2778	4.82	1.13	5	4.96	1.48	1	6	5	-0.90	0.43	0.02
O2	22	2800	2.71	1.57	2	2.56	1.48	1	6	5	0.59	-0.81	0.03
O3	23	2772	4.44	1.22	5	4.56	1.48	1	6	5	-0.77	0.30	0.02
O4	24	2786	4.89	1.22	5	5.10	1.48	1	6	5	-1.22	1.08	0.02
O5	25	2780	2.49	1.33	2	2.34	1.48	1	6	5	0.74	-0.24	0.03
gender	26	2800	1.67	0.47	2	1.71	0.00	1	2	1	-0.73	-1.47	0.01
education	27	2577	3.19	1.11	3	3.22	1.48	1	5	4	-0.05	-0.32	0.02
age	28	2800	28.78	11.13	26	27.43	10.38	3	86	83	1.02	0.56	0.21

How many factors?

```
> fa.parallel(bfi[1:25]) #just the items
```

Parallel analysis suggests that the number of factors = 6 and the number 6



How many factors part 2: VSS

```
> VSS(bfi[1:25])
```

Very Simple Structure

Call: VSS(x = bfi[1:25])

VSS complexity 1 achieves a maximum of 0.58 with 4 factors

VSS complexity 2 achieves a maximum of 0.74 with 4 factors

The Velicer MAP criterion achieves a minimum of 0.01 with 5 factors

Velicer MAP

```
[1] 0.02 0.02 0.02 0.02 0.01 0.02 0.02 0.02
```

Very Simple Structure Complexity 1

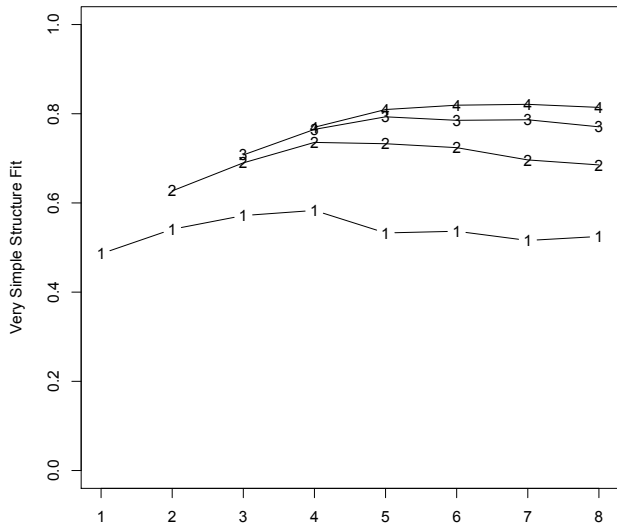
```
[1] 0.49 0.54 0.57 0.58 0.53 0.54 0.52 0.52
```

Very Simple Structure Complexity 2

```
[1] 0.00 0.63 0.69 0.74 0.73 0.72 0.70 0.69
```

VSS plot

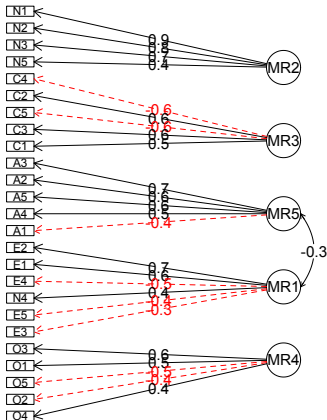
Very Simple Structure



Extract 5 factors from the BFI

```
> f5 <- fa(bfi[1:25],5)
fa.diagram(f5 ,main="Five_factors_of_personality?")
```

Five factors of personality?



ICLUST of Big 5

```
> iclust(bfi[1:25])
```

ICLUST (Item Cluster Analysis)

Purified Alpha:

C20	C16	C15	C21
0.80	0.81	0.73	0.61

G6* reliability:

C20	C16	C15	C21
0.82	0.81	0.72	0.61

Original Beta:

C20	C16	C15	C21
0.63	0.76	0.67	0.27

Cluster size:

C20	C16	C15	C21
10	5	5	5

With eigenvalues of:

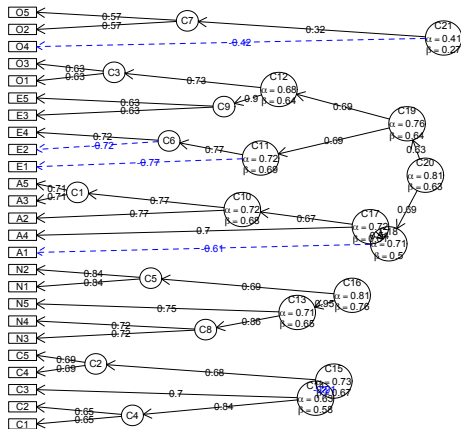
C20	C16	C15	C21
3.8	3.0	2.6	1.9

Purified **scale** intercorrelation reliabilities **on** diagonal correlations corrected **for**

	C20	C16	C15
C21			
C20	0.80	-0.29	-0.40
C16	-0.24	0.81	0.29
	0.11		
C15	-0.30	0.22	0.73
	0.30		
C21	-0.23	0.07	0.20
	0.61		

ICLUST as a graphic tree structure

Hierarchical Clusters of the Big 5



How many factors in the spi

R code

```
dim(spi)
nfactors(spi[11:145], n=40)
```

```
> dim(spi)
[1] 4000 145
```

```
> nfactors(spi[11:145], n=40)
```

Number of factors

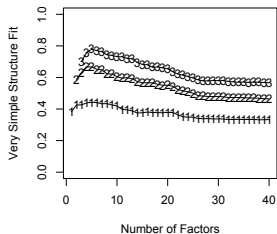
```
Call: vss(x = x, n = n, rotate = rotate, diagonal = diagonal, fm = fm,
  n.obs = n.obs, plot = FALSE, title = title, use = use, cor = cor)
VSS complexity 1 achieves a maximum of 0.44 with 5 factors
VSS complexity 2 achieves a maximum of 0.67 with 4 factors
The Velicer MAP achieves a minimum of 0 with 27 factors
Empirical BIC achieves a minimum of -39576.84 with 28 factors
Sample Size adjusted BIC achieves a minimum of -14717.22 with 39 factors
```

Statistics by number of factors

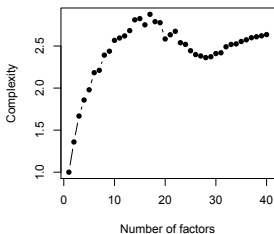
	vss1	vss2	map	dof	chisq	prob	sqresid	fit	RMSEA	BIC	SABIC	complex	eChisq	S
1	0.38	0.00	0.0192	8910	236125	0.0e+00	386	0.38	0.080	162225	190537	1.0	1108667	0.1
2	0.43	0.59	0.0141	8776	196282	0.0e+00	257	0.59	0.073	123493	151379	1.4	666677	0.0
3	0.43	0.63	0.0109	8643	165078	0.0e+00	187	0.70	0.067	93392	120856	1.7	432483	0.0
4	0.44	0.67	0.0081	8511	140161	0.0e+00	139	0.78	0.062	69571	96615	1.9	278119	0.0

How many factors of the spi

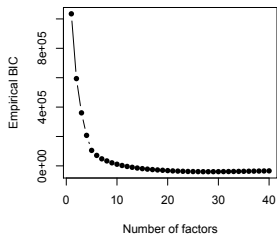
Very Simple Structure



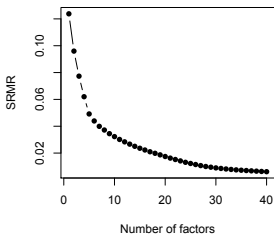
Complexity



Empirical BIC



Root Mean Residual



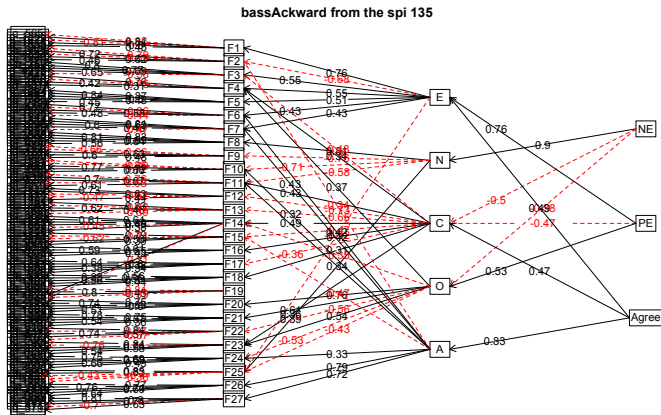
The bassAckwards algorithm

1. [Goldberg \(2006\)](#) suggested a simple hierarchical organization of component scores
2. [Waller \(2007\)](#) showed you did not need to actually find the scores, you could just use matrix algebra
3. `bassAckward` will do this on components or factors

R code

```
sp5 <- bassAckward(psychTools::spi[11:145], c(3,4,5,27))
```

spi135 bassAckward solution for 3, 5, and 27 factors



Analyzing from an external file

- Data may reside on a local file or on a remote computer
- Option A: Using read.clipboard and its alternatives
 - Open the other other file using a text editor or spreadsheet program
 - Select all and copy (to the clipboard)
 - `my.data <- read.clipboard()` or `my.data <- read.clipboard.csv()` or `read.clipboard.tab()`
- Read the information directly
 - find the file and call it something `fn <- file.choose()`
 - Read in the data `my.data <- read.table(fn, header=TRUE)`
- Read from an SPSS file using the foreign package
 - `library(foreign)`
 - find the file and call it something `fn <- file.choose()`
 - `my.data <- read.spss(fn, to.data.frame=TRUE)`
- Read the file directly, and translate as necessary
 - `my.data <- read.file()`
 - This will call `file.choose`, and then, depending upon the suffix, translate it

A simplex

- In developmental, or any time process, nearby items are more correlated
 - An underlying growth process
 - localized errors
- grades in progressive quarters
- reaction times during a long session

Simulate a simplex

```
> set.seed(42) # for reproducible results
> s9 <- sim.simplex(9,n=1000)
> str(s9) #show the structure
List of 4
 $ model      : num [1:9, 1:9] 1 0.8 0.64 0.512 0.41 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:9] "V1" "V2" "V3" "V4" ...
  .. ..$ : chr [1:9] "V1" "V2" "V3" "V4" ...
 $ r          : num [1:9, 1:9] 1 0.789 0.625 0.492 0.42 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:9] "V1" "V2" "V3" "V4" ...
  .. ..$ : chr [1:9] "V1" "V2" "V3" "V4" ...
 $ observed: num [1:1000, 1:9] -0.659 -0.858 0.241 0.714 1.268 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : NULL
  .. ..$ : chr [1:9] "V1" "V2" "V3" "V4" ...
 $ Call      : language sim.simplex(nvar = 9, n = 1000)
- attr(*, "class")= chr [1:2] "psych" "sim"
```

A simplex correlation matrix

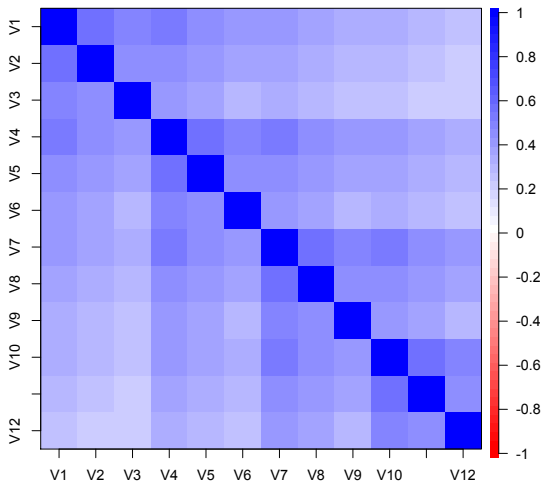
```
> round(s9$model,2)
```

	V1	V2	V3	V4	V5	V6	V7	V8	V9
V1	1.00	0.80	0.64	0.51	0.41	0.33	0.26	0.21	0.17
V2	0.80	1.00	0.80	0.64	0.51	0.41	0.33	0.26	0.21
V3	0.64	0.80	1.00	0.80	0.64	0.51	0.41	0.33	0.26
V4	0.51	0.64	0.80	1.00	0.80	0.64	0.51	0.41	0.33
V5	0.41	0.51	0.64	0.80	1.00	0.80	0.64	0.51	0.41
V6	0.33	0.41	0.51	0.64	0.80	1.00	0.80	0.64	0.51
V7	0.26	0.33	0.41	0.51	0.64	0.80	1.00	0.80	0.64
V8	0.21	0.26	0.33	0.41	0.51	0.64	0.80	1.00	0.80
V9	0.17	0.21	0.26	0.33	0.41	0.51	0.64	0.80	1.00

How many factors?

fa.parallel(s9\$observed)

Correlation plot



Factor a simplex

```
> f2 <- fa(s9$observed, 2)
```

```
> f2
```

Factor Analysis using method = minres

Call: fa(r = s9\$observed, nfactors = 2)

Standardized loadings based upon correlation **matrix**

	MR1	MR2	h2	u2
V1	-0.07	0.76	0.53	0.47
V2	-0.06	0.89	0.75	0.25
V3	-0.01	0.92	0.83	0.17
V4	0.21	0.72	0.71	0.29
V5	0.48	0.49	0.69	0.31
V6	0.72	0.23	0.74	0.26
V7	0.90	0.04	0.85	0.15
V8	0.91	-0.10	0.76	0.24
V9	0.79	-0.09	0.57	0.43

	MR1	MR2
SS loadings	3.22	3.19
Proportion Var	0.36	0.35
Cumulative Var	0.36	0.71

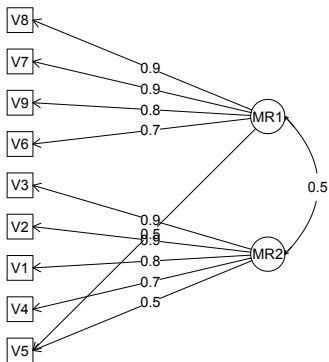
With **factor** correlations of

	MR1	MR2
MR1	1.00	0.47
MR2	0.47	1.00

factor diagram

> fa.diagram(f2,simple=FALSE) #show large cross loadings

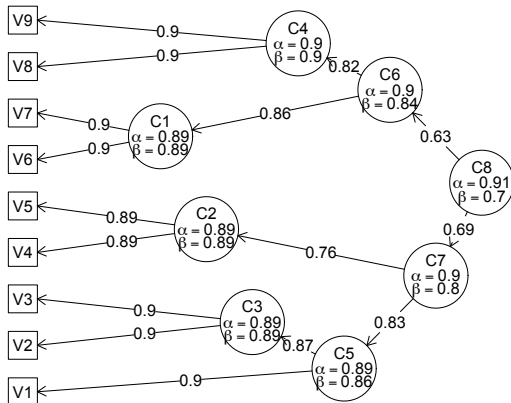
Factor Analysis



ICLUST of a simplex

> iclust(s9\$observed) #cluster analyze the data

iclust



Item difficulty leads to a simplex structure

- Dichotomous items (e.g., ability items) differ in difficulty
 - Easy items have high endorsement rates
 - Hard items have low endorsement rates
- Φ coefficient is sensitive to differences in response
- Items with similar difficulties will correlate more highly

How many factors

```
> set.seed(42)
> v9 <- sim.rasch(9)
> round(cor(v9$items), 2)
```

	V1	V2	V3	V4	V5	V6	V7	V8	V9
V1	1.00	0.11	0.01	0.12	0.06	0.09	0.11	0.03	0.06
V2	0.11	1.00	0.16	0.23	0.09	0.08	0.09	0.10	0.09
V3	0.01	0.16	1.00	0.14	0.07	0.17	0.10	0.08	0.07
V4	0.12	0.23	0.14	1.00	0.23	0.23	0.13	0.12	0.12
V5	0.06	0.09	0.07	0.23	1.00	0.21	0.07	0.11	0.06
V6	0.09	0.08	0.17	0.23	0.21	1.00	0.21	0.05	0.16
V7	0.11	0.09	0.10	0.13	0.07	0.21	1.00	0.12	0.09
V8	0.03	0.10	0.08	0.12	0.11	0.05	0.12	1.00	0.02
V9	0.06	0.09	0.07	0.12	0.06	0.16	0.09	0.02	1.00

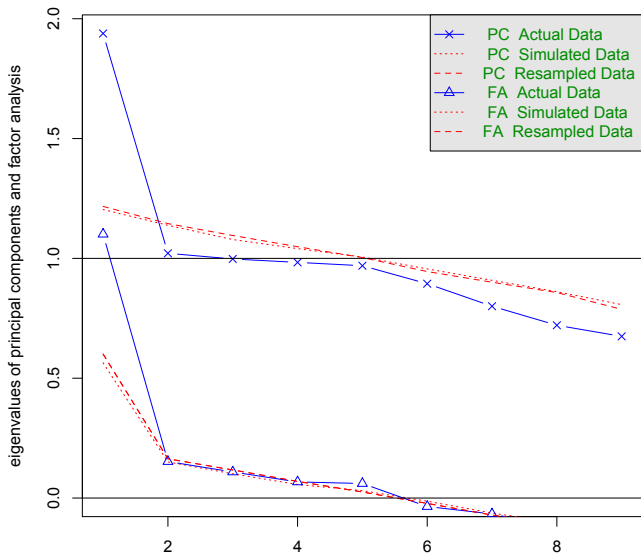
```
> fa.parallel(v9$items)
```

Parallel analysis suggests that the number of factors
5

and the number of components = 1

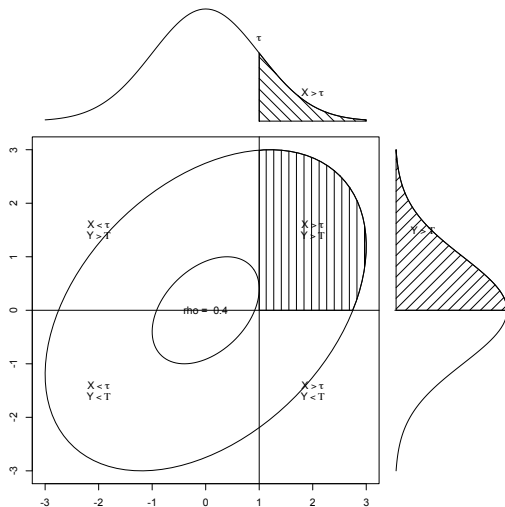
Parallel analysis of dichotomous items

Parallel Analysis Scree Plots



Find the tetrachoric correlations

```
> draw.tetra(.4,1,0) #rho, cut 1, cut 2
```



The tetrachoric correlation matrix

```
> rtet <- tetrachoric(v9$items)
```

```
Loading required package: mvtnorm
```

```
> rtet
```

```
Call: tetrachoric(x = v9$items)
```

```
tetrachoric correlation
```

	V1	V2	V3	V4	V5	V6	V7	V8	V9
V1	1.000	0.27	0.024	0.27	0.15	0.24	0.33	0.115	0.253
V2	0.268	1.00	0.299	0.41	0.18	0.15	0.21	0.281	0.322
V3	0.024	0.30	1.000	0.24	0.11	0.32	0.20	0.195	0.182
V4	0.273	0.41	0.242	1.00	0.36	0.38	0.23	0.253	0.288
V5	0.147	0.18	0.114	0.36	1.00	0.33	0.12	0.221	0.130
V6	0.239	0.15	0.316	0.38	0.33	1.00	0.35	0.111	0.335
V7	0.330	0.21	0.195	0.23	0.12	0.35	1.00	0.247	0.212
V8	0.115	0.28	0.195	0.25	0.22	0.11	0.25	1.000	0.048
V9	0.253	0.32	0.182	0.29	0.13	0.34	0.21	0.048	1.000

```
with tau of
```

	V1	V2	V3	V4	V5	V6	V7	V8	V9
	-1.46	-1.00	-0.72	-0.32	0.00	0.39	0.69	1.16	1.38

Factor analyze the items using tetrachorics

```
> f.irt <- irt.fa(v9$items)
> f.irt
```

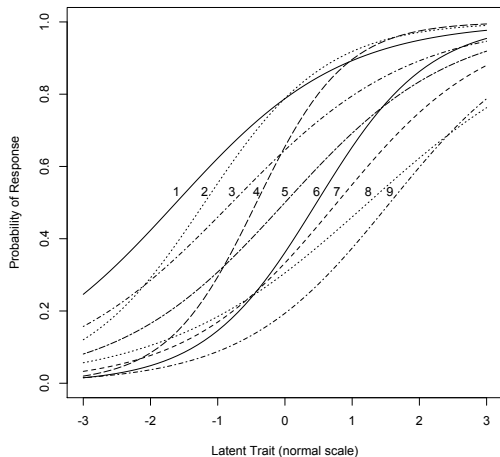
Item Response Analysis using Factor Analysis =

Call: irt.fa(x = v9\$items)

	Location	Discrimination	tau	Loading
V1	-1.62	0.48	-1.46	0.43
V2	-1.19	0.65	-1.00	0.54
V3	-0.79	0.45	-0.72	0.41
V4	-0.42	0.89	-0.32	0.66
V5	0.00	0.48	0.00	0.43
V6	0.47	0.71	0.39	0.58
V7	0.78	0.53	0.69	0.47
V8	1.24	0.39	1.16	0.36
V9	1.56	0.54	1.38	0.47

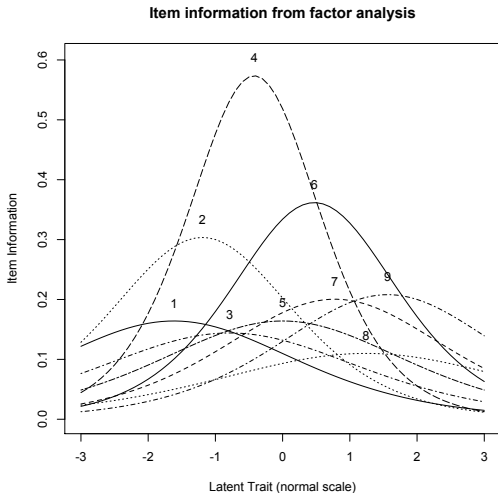
Show the items

Item parameters from factor analysis



```
> plot(f.irt,type="ICC")
```

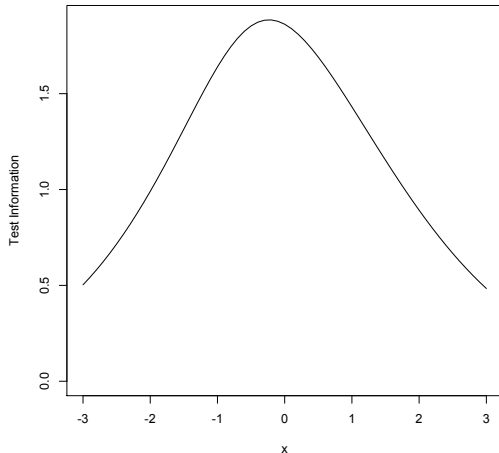
Show the item information functions



```
> plot(f.irt)
```

Show the Test information function

Test information -- item parameters from factor analysis



```
> plot(f.irt,type="test")
```

Polytomous items

- Most personality items have 3-6 alternatives
 - The fewer the alternatives, the more the correlation is restricted.
 - For 6 choice items this is not too serious, but for 4, it probably is.
- Find the polychoric correlation (What would be the Pearson if the data were bivariate normal?)
- polychoric function in R

Find polychoric correlations for Big 5 items

```
data(bfi)
rbfi <- polychoric(bfi[1:25]) #this takes awhile
#or Rbfi <- mixed.cor(bfi) #find polychorics, biserials, and polyserials
```

MDS to remove a general factor

When working with variables sharing a large general factor such as general intelligence in the cognitive domain, or neuroticism in the clinical domain, all the entries in the correlation matrix will be large and thus, the average correlation will be large. Especially if orthogonal rotations are used, it is difficult to see the structure of the smaller factors. A factor analytic solution is to use oblique transformations, extract second order factors, and then plot the loadings on the lower level factors. An alternative to this is to represent the correlations in terms of deviations from the average correlation. One way to do this is to convert the *correlations* to *distances* and do a *multidimensional scaling* of the distances. As was shown earlier *distance* between two variables is an inverse function of their correlation:

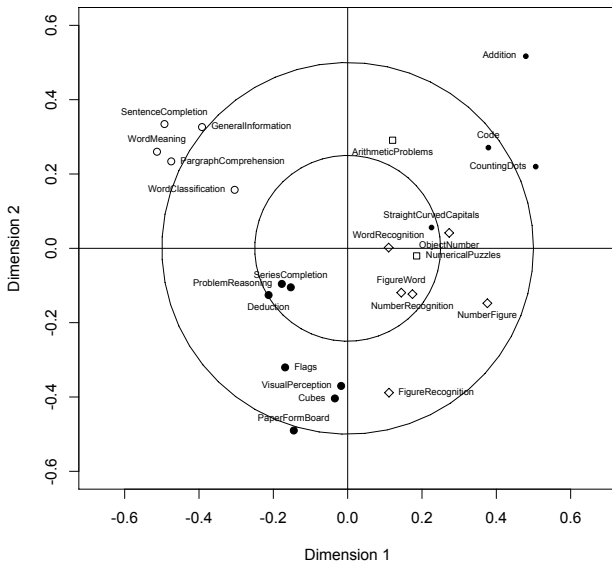
$$d_{xy} = \sqrt{2 * (1 - r_{xy})}. \quad (25)$$

Code to show MDS solution

Table: Multidimensional scaling of 24 mental tests. See Figure 1 for a graphical representation. The correlations are transformed to distances using Equation 25. The multidimensional scaling uses the `cmdscale` function.

```
> dis24 <- sqrt(2*(1-Harman74.cor$cov))
> mds24 <- cmdscale(dis24,2)
> plot.char <- c( 19, 19, 19, 19, 21, 21, 21, 21, 21, 20, 20, 20,
  20, 23, 23, 23, 23, 23, 23, 19, 22, 19, 19, 22 )
> plot(mds24, xlim=c(-.6,.6), ylim=c(-.6,.6), xlab="Dimension_1", ylab="Dimension_2", asp=1, pch=plot.char)
> position <- c(2,2,3,4, 4,4,3,4, 3,2,3,2, 3,3,1,4, 4,1,3,1, 1,2,3,4)
> text(mds24, rownames(mds24), cex=.6, pos=position)
> abline(v=0,h=0)
> title("Multidimensional_Scaling_of_24_ability_tests")
> #draw circles at .25 and .50 units away from the center
> segments = 51
> angles <- (0:segments) * 2 * pi/segments
> unit.circle <- cbind(cos(angles), sin(angles))
> lines(unit.circle*.25)
> lines(unit.circle*.5)
```

Multidimensional Scaling of 24 ability tests



Non-hiearcical clustering

1. Find the proximity (e.g. correlation) matrix,
2. Identify the most similar triplet of items,
3. Combine this most similar triplet of items to form a seed cluster,
4. Add items to this cluster that increase the cluster saturation,
5. Eliminate items that would reduce the cluster saturation,
6. Repeat steps 4 and 5 until no items remain. This is a homogeneous subtest.
7. Identify a new nucleus set of three items from the remaining items and start at step 3.
8. Repeat steps 4-7 until no items remain. The clusters are both maximally homogeneous and maximally independent.

Hierarchical Clustering (e.g., iclust)

1. Find the proximity (e.g. correlation) matrix,
2. Identify the most similar pair of items
3. Combine this most similar pair of items to form a new variable (cluster),
4. Find the similarity of this cluster to all other items and clusters,
5. Repeat steps 2 and 3 until some criterion is reached (e.g., typically, if only one cluster remains or in ICLUST if there is a failure to increase reliability coefficients α or β).
6. Purify the solution by reassigning items to the most similar cluster center.

ICLUST of 24 mental abilities



Factor score estimates are estimates, not precise

That factor scores are indeterminate has been taken by some (e.g., [Schonemann, 1996](#)) to represent psychopathology on the part of psychometricians, for the problem of indeterminacy has been known (and either ignored or suppressed) since [Wilson \(1928\)](#).

To others, factor indeterminacy is a problem at the data level but not the structural level, and at the data level it is adequate to report the degree of indeterminacy.

This degree of indeterminacy is indeed striking ([Schonemann and Wang, 1972](#); [Velicer and Jackson, 1990](#)), and should be reported routinely. It is reported for each factor in the `fa` and `omega` functions.

Factor Extension

Table: Create 12 variables with a clear two factor structure. Remove variables 3, 6, 9, and 12 from the matrix and factor the other variables. Then extend this solution to the deleted variables. Compare this solution to the solution with all variables included (not shown). A graphic representation is in Figure 2.

```
set.seed(42)
fx <- matrix(c(.9,.8,.7,.85,.75,.65,rep(0,12),.9,.8,.7,.85,.75,.65),ncol=2)
Phi <- matrix(c(1,.6,.6,1),2)
sim.data <- sim.structure(fx,Phi,n=1000,raw=TRUE)
R <- cor(sim.data$observed)
Ro <- R[c(1,2,4,5,7,8,10,11),c(1,2,4,5,7,8,10,11)]
Roe <- R[c(1,2,4,5,7,8,10,11),c(3,6,9,12)]
fo <- fa(Ro,2)
fe <- fa.extension(Roe,fo)
fa.diagram(fo,fe=fe)
```

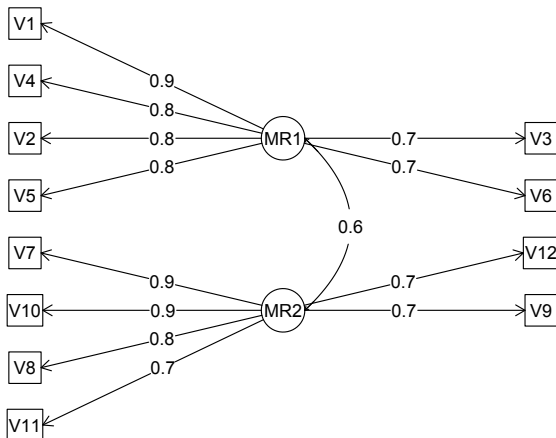
Call: fa.extension(Roe = Roe, fo = fo)

Standardized loadings based upon correlation matrix

	MR1	MR2	h2	u2
V3	0.69	0.01	0.49	0.51
V6	0.66	-0.02	0.42	0.58
V9	0.01	0.66	0.44	0.56
V12	-0.06	0.70	0.44	0.56

	MR1	MR2
SS loadings	0.89	0.89

Factor analysis and extension



Anderson, T. and Rubin, H. (1956). Statistical inference in factor analysis. *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability, Volume 5: Contributions to Econometrics, Industrial Research, and Psychometry*, 5:111–150.

Bartlett, M. S. (1937). The statistical conception of mental factors. *British Journal of Psychology. General Section*, 28(1):97–104.

Bartlett, M. S. (1951). The effect of standardization on a χ^2 approximation in factor analysis. *Biometrika*, 38(3/4):337–344.

Burt, C. (1948). The factorial study of temperamental traits. *British Journal of Statistical Psychology*, 1(3):178–203.

Carroll, J. (1953). An analytical solution for approximating simple structure in factor analysis. *Psychometrika*, 18(1):23–38.

Cattell, R. B. (1965). A biometrics invited paper. factor analysis: An introduction to essentials i. the purpose and underlying models. *Biometrics*, 21(1):190–215.

Cattell, R. B. (1966). The scree test for the number of factors. *Multivariate Behavioral Research*, 1(2):245–276.

Condon, D. M. (2018). The SAPA Personality Inventory: An empirically-derived, hierarchically-organized self-report personality assessment model. *PsyArXiv*.

Eckart, C. and Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218.

Ferguson, G. (1954). The concept of parsimony in factor analysis. *Psychometrika*, 19(4):281–290.

Goldberg, L. R. (2006). Doing it all Bass-Ackwards: The development of hierarchical factor structures from the top down. *Journal of Research in Personality*, 40(4):347 – 358.

Gorsuch, R. L. (1983). *Factor analysis*. L. Erlbaum Associates, Hillsdale, N.J., 2nd edition.

Grice, J. W. (2001). Computing and evaluating factor scores. *Psychological Methods*, 6(4):430–450.

Harman, H. H. (1976). *Modern factor analysis*. University of Chicago Press, Chicago, 3d ed., rev edition.

Horn, J. L. (1965). A rationale and test for the number of factors in factor analysis. *Psychometrika*, 30(2):179–185.

Householder, A. S. and Young, G. (1938). Matrix approximation and latent roots. *The American Mathematical Monthly*, 45(3):165–171.

Jennrich, R. (2001). A simple general procedure for orthogonal rotation. *Psychometrika*, 66(2):289–306.

Jennrich, R. (2002). A simple general method for oblique rotation. *Psychometrika*, 67(1):7–19.

Jennrich, R. (2004). Derivative free gradient projection algorithms for rotation. *Psychometrika*, 69(3):475–480.

Jöreskog, K. G. (1978). Structural analysis of covariance and correlation matrices. *Psychometrika*, 43(4):443–477.

Kaiser, H. F. (1958). The varimax criterion for analytic rotation in factor analysis. *Psychometrika*, 23(3):187–200.

Lawley, D. N. and Maxwell, A. E. (1962). Factor analysis as a statistical method. *The Statistician*, 12(3):209–229.

Lawley, D. N. and Maxwell, A. E. (1963). *Factor analysis as a statistical method*. Butterworths, London.

McDonald, R. P. (1981). The dimensionality of tests and items. *British Journal of mathematical and statistical Psychology*, 34(1):100–117.

Neuhaus, J. and Wrigley, C. (1954). The quartimax method: an analytical approach to orthogonal simple structure. *British Journal of Statistical Psychology*, 7:81–91.

Revelle, W. and Rocklin, T. (1979). Very Simple Structure - alternative procedure for estimating the optimal number of interpretable factors. *Multivariate Behavioral Research*, 14(4):403–414.

Roff, M. (1936). Some properties of the communality in multiple factor theory. *Psychometrika*, 1(2):1–6.

Schonemann, P. H. (1996). The psychopathology of factor indeterminacy. *Multivariate Behavioral Research*, 31(4):571–577.

Schonemann, P. H. and Wang, M.-M. (1972). Some new results on factor indeterminacy. *Psychometrika*, . 37(1, Pt. 1):61–91.

Spearman, C. (1904). "General Intelligence," objectively determined and measured. *American Journal of Psychology*, 15(2):201–292.

ten Berge, J. M., Krijnen, W. P., Wansbeek, T., and Shapiro, A. (1999). Some new results on correlation-preserving factor scores prediction methods. *Linear Algebra and its Applications*, 289(1-3):311 – 318.

Thurstone, L. L. (1947). *Multiple-factor analysis: a development and expansion of The vectors of the mind*. The University of Chicago Press, Chicago, Ill.

Velicer, W. (1976). Determining the number of components from the matrix of partial correlations. *Psychometrika*, 41(3):321–327.

Velicer, W. F. and Jackson, D. N. (1990). Component analysis versus common factor analysis: Some issues in selecting an appropriate procedure. *Multivariate Behavioral Research*, 25(1):1–28.

Waller, N. (2007). A general method for computing hierarchical component structures by Goldberg's Bass-Ackwards method. *Journal of Research in Personality*, 41(4):745 – 752.

Wilson, E. B. (1928). On hierarchical correlation systems. *Proceedings of the National Academy of Sciences*, 14(3):283–291.