# An introduction to R: a short course Part II

Multivariate analysis

# The zelig website

**ZELIG: EVERYONE'S STATISTICAL SOFTWARE**

Kosuke Imai, Gary King, Olivia Lau

**Version:3.4-5** (Download) (What's new)


Zelig is named after a Woody Allen movie about a man who had the strange ability to become the physical and psychological reflection of anyone he met and thus to fit perfectly in any situation.

Zelig is a single, easy-to-use program that can estimate, help interpret, and present the results of a large range of statistical methods. It literally *is* "everyone's statistical software" because Zelig uses (R) code from many researchers. We also hope it will *become* "everyone's statistical software" for applications, and we have designed it so that anyone can use it or add their methods to it. Zelig comes with detailed, self-contained documentation that minimizes startup costs for Zelig and R, automates graphics and summaries for all models, and, with only three simple commands required, generally makes the power of R accessible for all users. Zelig also works well for teaching, and is designed so that scholars can use the same program with students that they use for their research.

Zelig adds considerable infrastructure to improve the use of existing methods. It generalizes the program Clarify (for Stata), which translates hard-to-interpret coefficients into quantities of interest; combines multiply imputed data sets (such as output from Amelia) to deal with missing data; automates bootstrapping for all models; uses sophisticated nonparametric matching commands which improve parametric procedures (via MatchIt); allows one-line commands to run analyses in all designated strata; automates the creation of replication data files so that you (or, if you wish, anyone else) can replicate the results of your analyses (hence satisfying the replication standard); makes it easy to evaluate counterfactuals (via WhatIf); and allows conditional population and superpopulation inferences. Zelig includes many specific methods, based on likelihood, frequentist, Bayesian, robust Bayesian, and nonparametric theories of inference.

- Documentation and Installation
- Installation, Frequently Asked Questions
- Please send **All** questions, bugs and requests: Zelig Mailing List, [Un]Subscribe, or Browse/Search Archives
- A paper that describes the advances underlying Zelig software: Kosuke Imai, Gary King, and Olivia Lau. **"Toward A Common Framework for Statistical Analysis and Development"** *Journal of Computational and Graphical Statistics*, Vol. 17, No. 4 (December), pp. 892-913 (**Abstract:** HTML | **Paper:** PDF)
- Slides used to introduce Zelig
- Be notified of Zelig changes

http://gking.harvard.edu/zelig/

# Multivariate analysis

I. Dimension reduction through factor analysis, principal components analysis, cluster analysis, and multidimensional scaling

II. Multiple measures of reliability

III. Practical use of R for scoring inventories

# Dimension Reduction

I. The problem: How best to summarize and think about many variables some of which are moderately correlated

II. The solutions: rank reduction through FA, PCA, CA, MDS

III. Examples will be tests and then items

# The Thurstone data set

```
> data(bifactor)
> colnames(Thurstone) <- c
("Sentences","Vocab","S.comp","F.letter","4.letter","Suffix",
     "Series","Pedi","letters")
```

```
> round(Thurstone,2)
                Sentences Vocab S.comp F.letter 4.letter Suffix Series Pedi letters
Sentences            1.00  0.83   0.78     0.44     0.43   0.45   0.45 0.54    0.38
Vocabulary           0.83  1.00   0.78     0.49     0.46   0.49   0.43 0.54    0.36
Sent.Completion      0.78  0.78   1.00     0.46     0.42   0.44   0.40 0.53    0.36
First.Letters        0.44  0.49   0.46     1.00     0.67   0.59   0.38 0.35    0.42
4.Letter.Words       0.43  0.46   0.42     0.67     1.00   0.54   0.40 0.37    0.45
Suffixes             0.45  0.49   0.44     0.59     0.54   1.00   0.29 0.32    0.32
Letter.Series        0.45  0.43   0.40     0.38     0.40   0.29   1.00 0.56    0.60
Pedigrees            0.54  0.54   0.53     0.35     0.37   0.32   0.56 1.00    0.45
Letter.Group         0.38  0.36   0.36     0.42     0.45   0.32   0.60 0.45    1.00
```

# How many dimensions

I. Chi Square test

II. Scree plot

III. Parallel analysis of random data

IV. Minimum Average Partial correlation

V. Very Simple Structure

VI. Do not use eigen value > 1 rule!

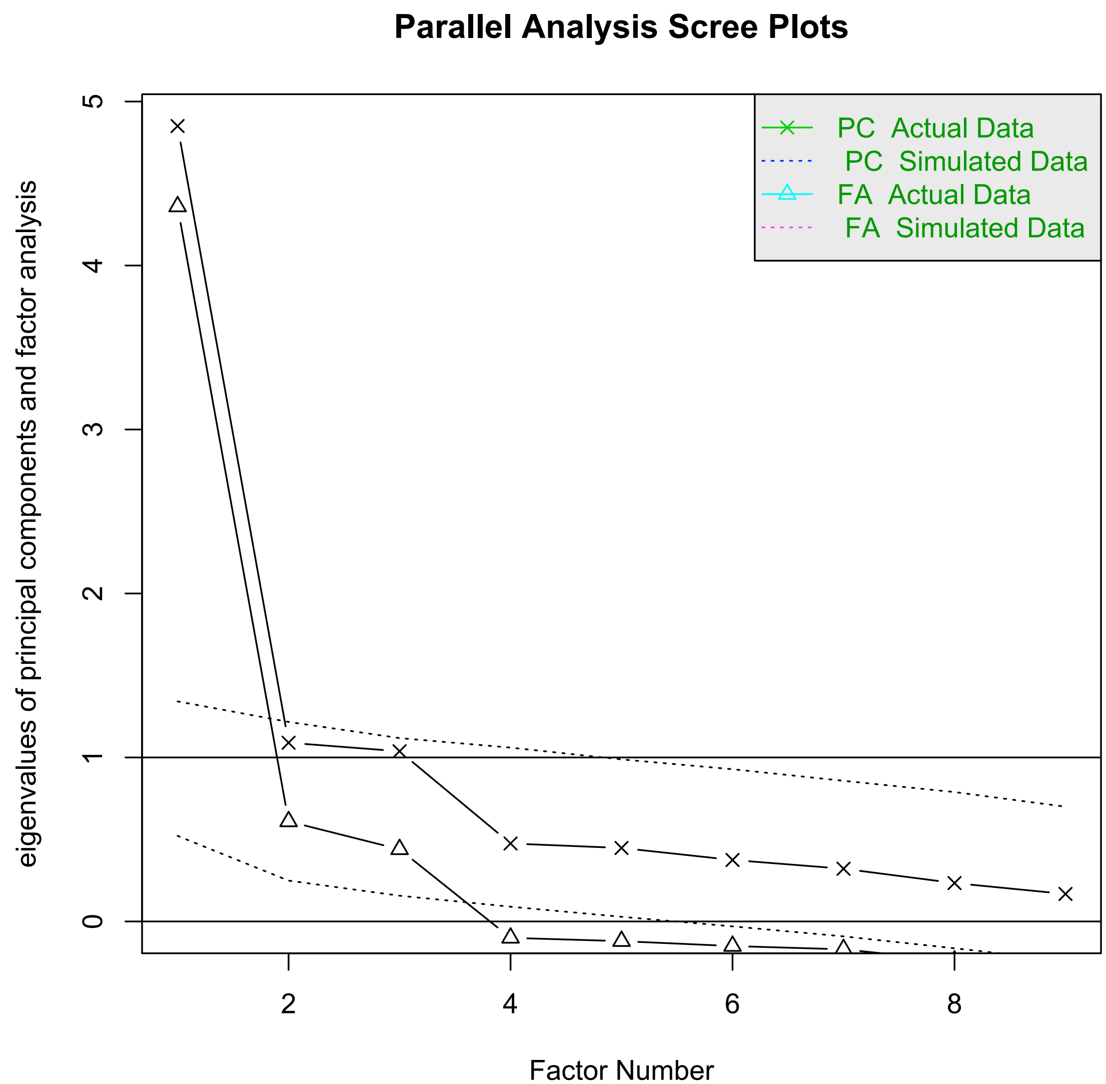
# MLE factor analyis: factanal

```
> factanal(covmat=Thurstone,factors=3,n.obs =213)
Call:
factanal(factors = 3, covmat = Thurstone, n.obs = 213)
Uniquenesses:
Sentences     Vocab    S.comp  F.letter  4.letter    Suffix    Series
Pedi   letters
    0.175     0.165     0.268     0.268     0.372     0.504     0.282
0.496     0.473
Loadings:
               Factor1 Factor2 Factor3
Sentences        0.834   0.244   0.264
Vocabulary       0.827   0.318   0.223
Sent.Completion 0.775   0.284   0.227
First.Letters    0.228   0.792   0.230
4.Letter.Words  0.213   0.706   0.291
Suffixes         0.314   0.616   0.134
Letter.Series    0.232   0.179   0.795
Pedigrees        0.446   0.166   0.527
Letter.Group     0.154   0.311   0.638
               Factor1 Factor2 Factor3
SS loadings       2.454   1.902   1.642
Proportion Var    0.273   0.211   0.182
Cumulative Var    0.273   0.484   0.666
Test of the hypothesis that 3 factors are sufficient.
The chi square statistic is 2.82 on 12 degrees of freedom.
The p-value is 0.997
```

```
Call:
factanal(factors = 2, covmat = Thurstone, n.obs = 213)
Uniquenesses:
Sentences      Vocab     S.comp  F.letter  4.letter    Suffix    Series
Pedi   letters
    0.168      0.178      0.269      0.322      0.344      0.537      0.680
0.612     0.677
Loadings:
                Factor1 Factor2
Sentences        0.866   0.287
Vocabulary       0.839   0.343
Sent.Completion 0.795   0.314
First.Letters    0.255   0.783
4.Letter.Words   0.235   0.775
Suffixes         0.317   0.602
Letter.Series    0.372   0.426
Pedigrees        0.524   0.336
Letter.Group     0.269   0.501
               Factor1 Factor2
SS loadings      2.793   2.420
Proportion Var   0.310   0.269
Cumulative Var   0.310   0.579
Test of the hypothesis that 2 factors are sufficient.
The chi square statistic is 82.84 on 19 degrees of freedom.
The p-value is 5.99e-10
```

chi square rejects a 2 factor solution

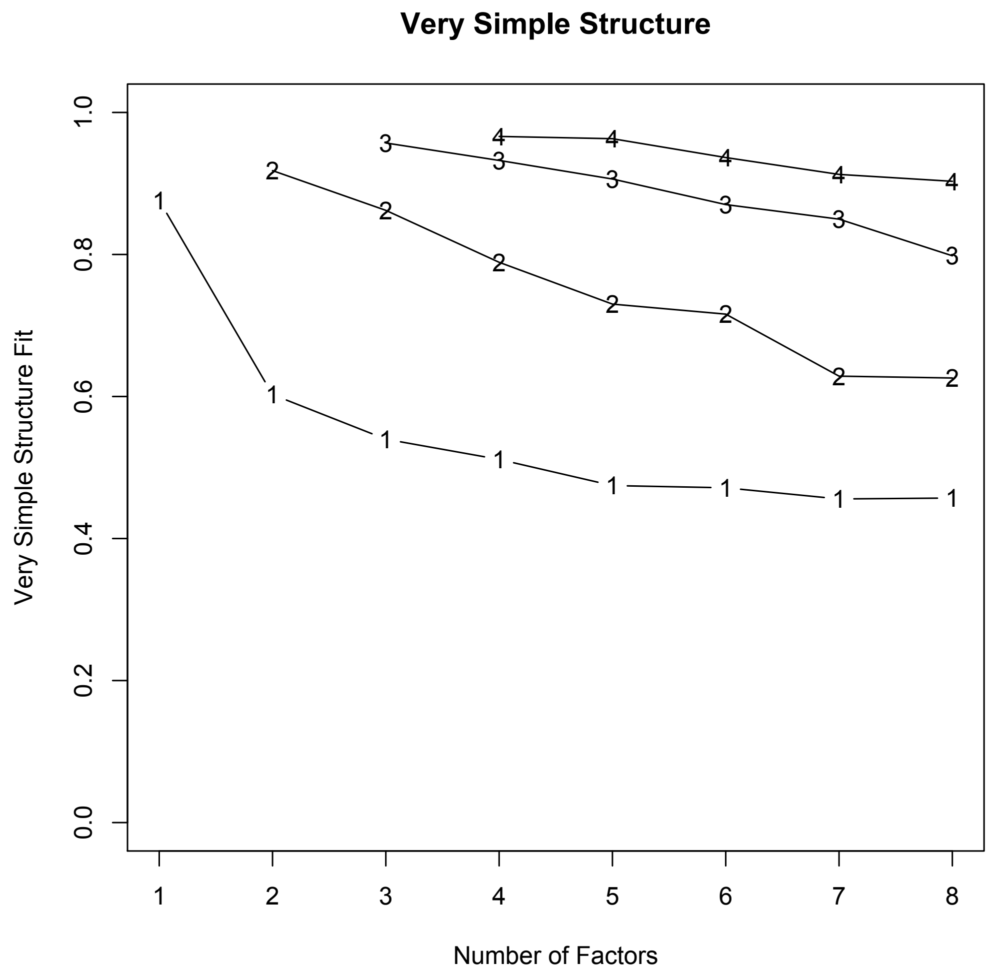
# Parallel analysis



# Very Simple Structure and MAP

```
> vss <- VSS(Thurstone,n.obs=213,SMC=FALSE)
> vss

Very Simple Structure
Call: VSS(x = Thurstone, n.obs = 213, SMC = FALSE)
VSS complexity 1 achieves a maximimum of 0.88  with  1  factors
VSS complexity 2 achieves a maximimum of 0.92  with  2  factors

The Velicer MAP criterion achieves a minimum of 1  with  3  factors

Velicer MAP
[1] 0.07 0.07 0.07 0.11 0.20 0.31 0.59 1.00

Very Simple Structure Complexity 1
[1] 0.88 0.60 0.54 0.51 0.47 0.47 0.46 0.46

Very Simple Structure Complexity 2
[1] 0.00 0.92 0.86 0.79 0.73 0.72 0.63 0.63
```

# Very Simple Structure



# Principal Axis FA

```
> pa3 <- factor.pa(Thurstone,nfactors=3,n.obs=213)
> pa3
          V  PA1  PA2  PA3
Sentences 1 0.83
Vocab     2 0.83 0.32
S.comp    3 0.78
F.letter  4      0.79
4.letter  5      0.71
Suffix    6 0.31 0.62
Series    7           0.79
Pedi      8 0.45      0.53
letters   9      0.31 0.64

                PA1  PA2  PA3
SS loadings    2.46 1.91 1.64
Proportion Var 0.27 0.21 0.18
Cumulative Var 0.27 0.49 0.67

Test of the hypothesis that 3 factors are sufficient.

The degrees of freedom for the model is 12  and the fit was  0.01
The number of observations was  213  with Chi Square =  2.97  with
prob <  1
```

# Factor analysis options

I. factanal (MLE) is part of core R

II. fa is part of psych

   A. minres (ols) (default)

   B. weighted least squares

   C. generalized weighted least squares

   D. maximum likelihood

   E. principal axis

# Principal Components

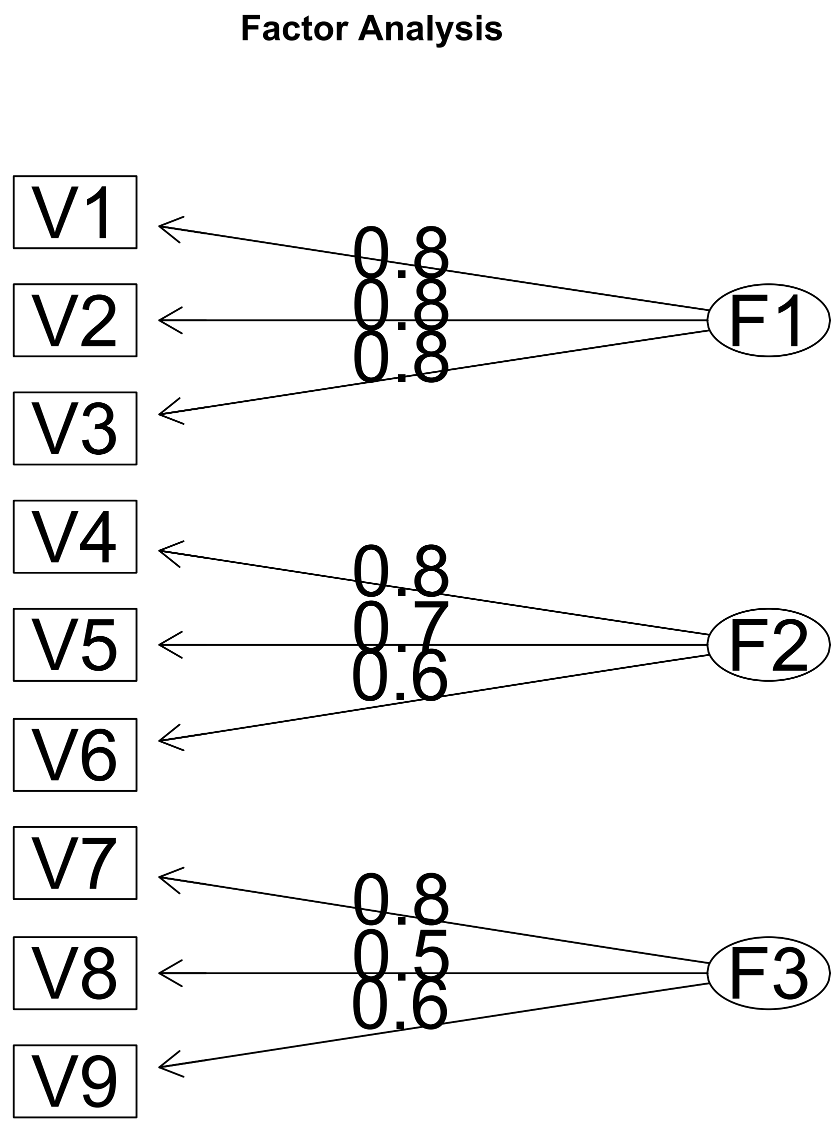
```
> pc3
                V   PC1   PC2   PC3
Sentences       1 0.863
Vocabulary      2 0.854  0.31
Sent.Completion 3 0.849
First.Letters   4        0.82
4.Letter.Words  5        0.79 0.301
Suffixes        6 0.314  0.77
Letter.Series   7             0.834
Pedigrees       8 0.534       0.613
Letter.Group    9        0.31 0.805

                PC1  PC2  PC3
SS loadings    2.73 2.25 1.99
Proportion Var 0.30 0.25 0.22
Cumulative Var 0.30 0.55 0.78

Test of the hypothesis that 3 factors are sufficient.

The degrees of freedom for the model is 12  and the fit was  0.62
The number of observations was  213  with Chi Square =  127.9  with
prob <  1.6e-21
```

# Comparing solutions: factor congruence

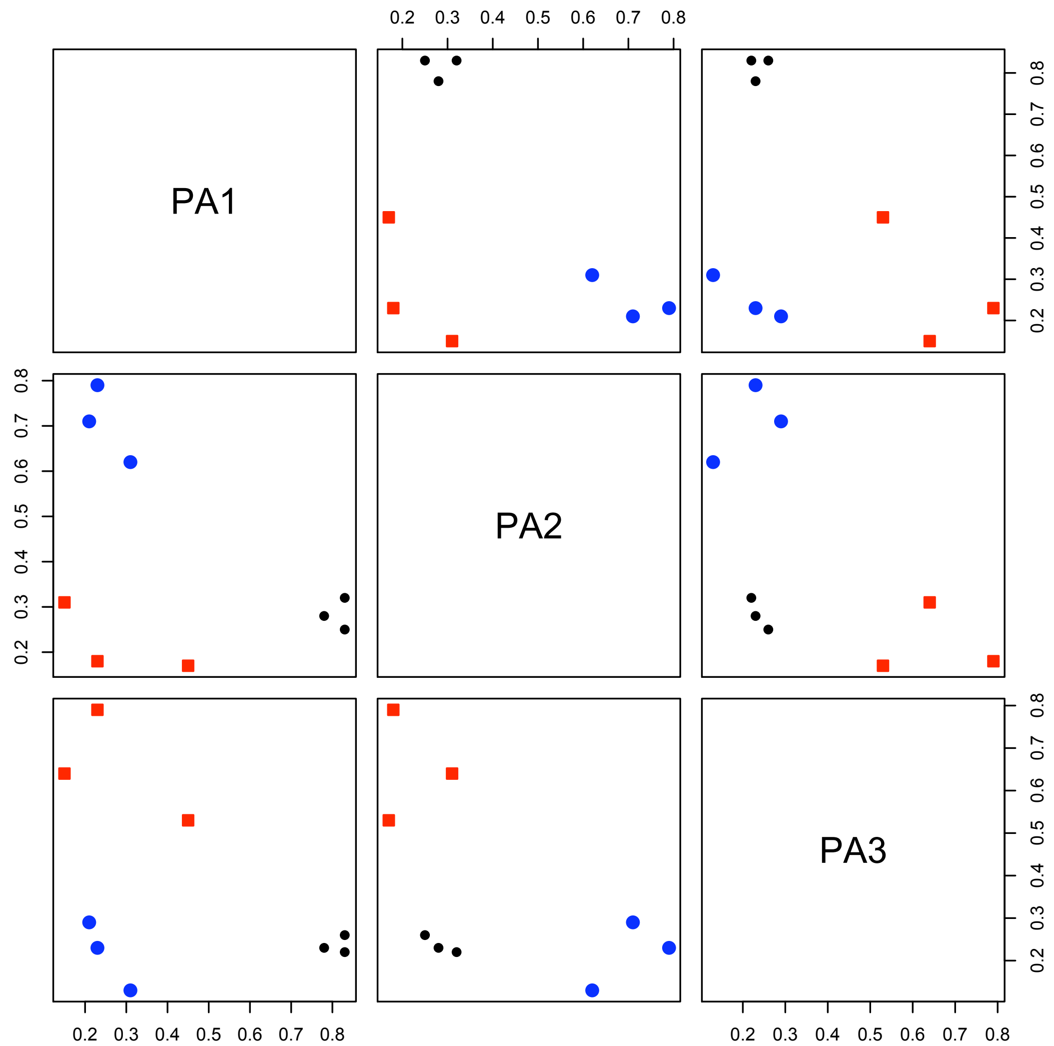
```
> factor.congruence(list(f3,pa3,pc3))
```

| | Factor1 | Factor2 | Factor3 | PA1 | PA2 | PA3 | PC1 | PC2 | PC3 |
|---|---|---|---|---|---|---|---|---|---|
| Factor1 | 1.00 | 0.64 | 0.62 | 1.00 | 0.64 | 0.62 | 1.00 | 0.59 | 0.55 |
| Factor2 | 0.64 | 1.00 | 0.62 | 0.63 | 1.00 | 0.62 | 0.61 | 0.99 | 0.57 |
| Factor3 | 0.62 | 0.62 | 1.00 | 0.62 | 0.62 | 1.00 | 0.61 | 0.56 | 0.99 |
| PA1 | 1.00 | 0.63 | 0.62 | 1.00 | 0.64 | 0.62 | 1.00 | 0.59 | 0.55 |
| PA2 | 0.64 | 1.00 | 0.62 | 0.64 | 1.00 | 0.62 | 0.61 | 0.99 | 0.57 |
| PA3 | 0.62 | 0.62 | 1.00 | 0.62 | 0.62 | 1.00 | 0.61 | 0.56 | 0.99 |
| PC1 | 1.00 | 0.61 | 0.61 | 1.00 | 0.61 | 0.61 | 1.00 | 0.56 | 0.54 |
| PC2 | 0.59 | 0.99 | 0.56 | 0.59 | 0.99 | 0.56 | 0.56 | 1.00 | 0.50 |
| PC3 | 0.55 | 0.57 | 0.99 | 0.55 | 0.57 | 0.99 | 0.54 | 0.50 | 1.00 |

# A misleading graph



# Plot the loadings: shows some cross loadings




`plot(pa3)`

# Rotations and transformations

I. Orthogonal rotations

   A. Varimax, Quartimax

II. Oblique transformations

   A. Promax, Oblimin, Quartimin, biquartimin, targeted, cluster, ...

```
> fa3o <- fa(Thurstone,3,rotate="oblimin")
Loading required package: GPArotation
> fa3o
Factor Analysis using method =  minres
Call: fa(r = Thurstone, nfactors = 3, rotate = "oblimin")
                item   MR1   MR2   MR3   h2   u2
Sentences          1  0.90             0.82 0.18
Vocabulary         2  0.89             0.84 0.16
Sent.Completion    3  0.84             0.74 0.26
First.Letters      4        0.85       0.73 0.27
4.Letter.Words     5        0.75       0.63 0.37
Suffixes           6        0.63       0.50 0.50
Letter.Series      7              0.84 0.72 0.28
Pedigrees          8  0.38        0.47 0.50 0.50
Letter.Group       9              0.63 0.52 0.48
                MR1  MR2  MR3
SS loadings    2.64 1.87 1.49
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 0.53
MR2 0.59 1.00 0.52
MR3 0.53 0.52 1.00
Test of the hypothesis that 3 factors are sufficient.
The degrees of freedom for the model is 12  and the objective function
```

# Oblique solution Oblimin

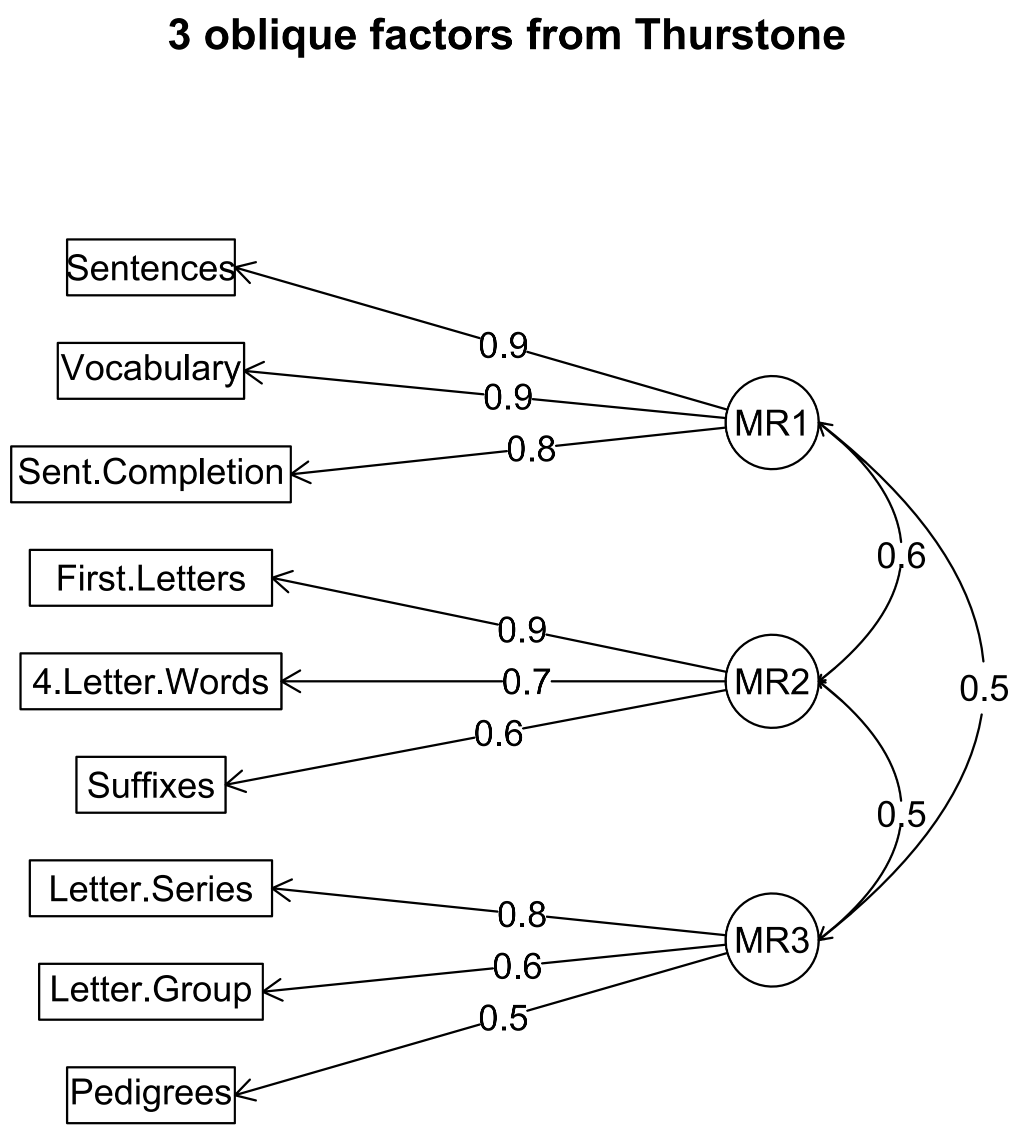
```
> print(fa3o,cut=0)
Factor Analysis using method =  minres
Call: fa(r = Thurstone, nfactors = 3, rotate = "oblimin")
                item   MR1   MR2   MR3   h2   u2
Sentences          1  0.90 -0.04  0.04 0.82 0.18
Vocabulary         2  0.89  0.06 -0.03 0.84 0.16
Sent.Completion    3  0.84  0.04  0.00 0.74 0.26
First.Letters      4  0.00  0.85  0.00 0.73 0.27
4.Letter.Words     5 -0.02  0.75  0.10 0.63 0.37
Suffixes           6  0.18  0.63 -0.08 0.50 0.50
Letter.Series      7  0.03 -0.01  0.84 0.72 0.28
Pedigrees          8  0.38 -0.05  0.47 0.50 0.50
Letter.Group       9 -0.06  0.21  0.63 0.52 0.48

                MR1  MR2  MR3
SS loadings    2.64 1.87 1.49
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 0.53
MR2 0.59 1.00 0.52
MR3 0.53 0.52 1.00
```

Show all values!

# Oblique: Promax

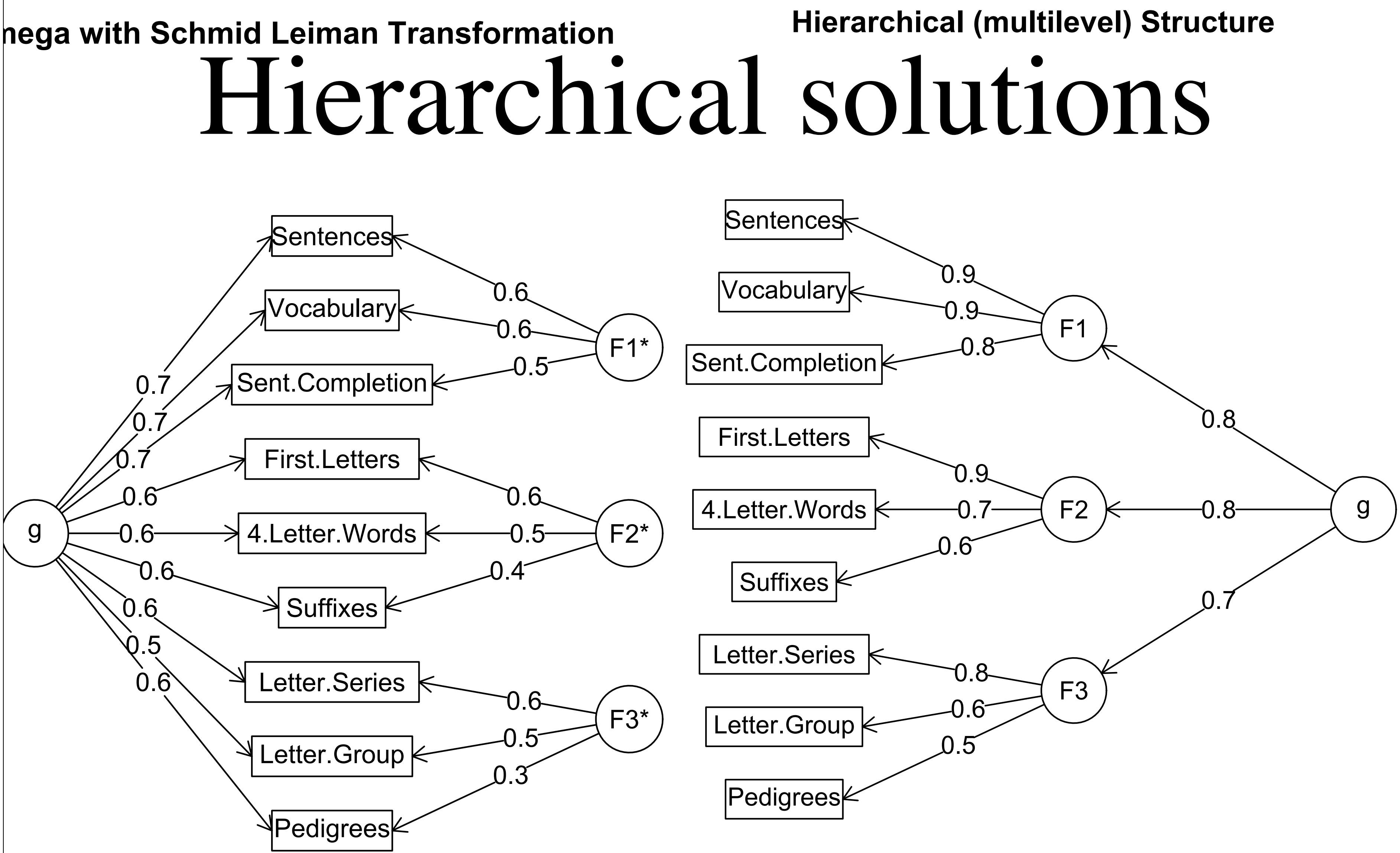
```
> pa3p <- Promax(pa3)
> pa3p
          V    PA1     PA2      PA3
Sentences 1  0.911
Vocab     2  0.904
S.comp    3  0.848
F.letter  4         0.869
4.letter  5         0.759
Suffix    6         0.650
Series    7                  0.8865
Pedi      8  0.350           0.4969
letters   9                  0.6800

                PA1  PA2  PA3
SS loadings    2.54 1.80 1.52
Proportion Var 0.28 0.20 0.17
Cumulative Var 0.28 0.48 0.65

 With factor correlations of
     PA1  PA2  PA3
PA1 1.00 0.61 0.61
PA2 0.61 1.00 0.58
PA3 0.61 0.58 1.00
```

# A more accurate graphic




```
fa.diagram(fa3o)
```

# Hierarchical solutions




```
> omsl <-omega(Thurstone,title="Bifactor")
> omsl <-omega(Thurstone,sl=FALSE,title="Hierarchical")
```

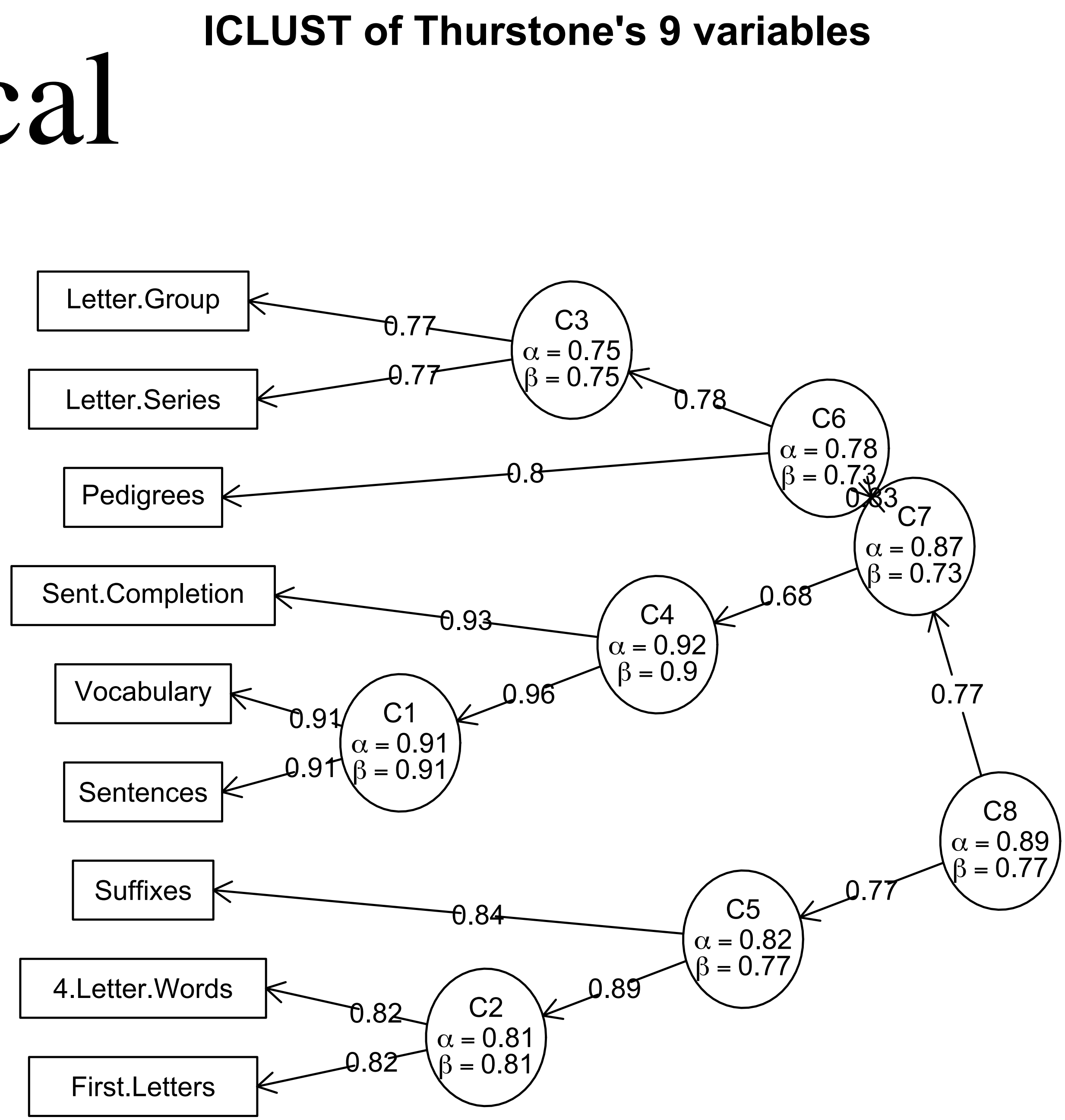
# Hierarchical Clustering

I. Find the similarity matrix (correlations)

II. Find the most similar pair of items/tests

III. Combine them and repeat II and III until some criterion (alpha, beta) fails to increase

# Hierarchical cluster analysis



# Multidimensional scaling

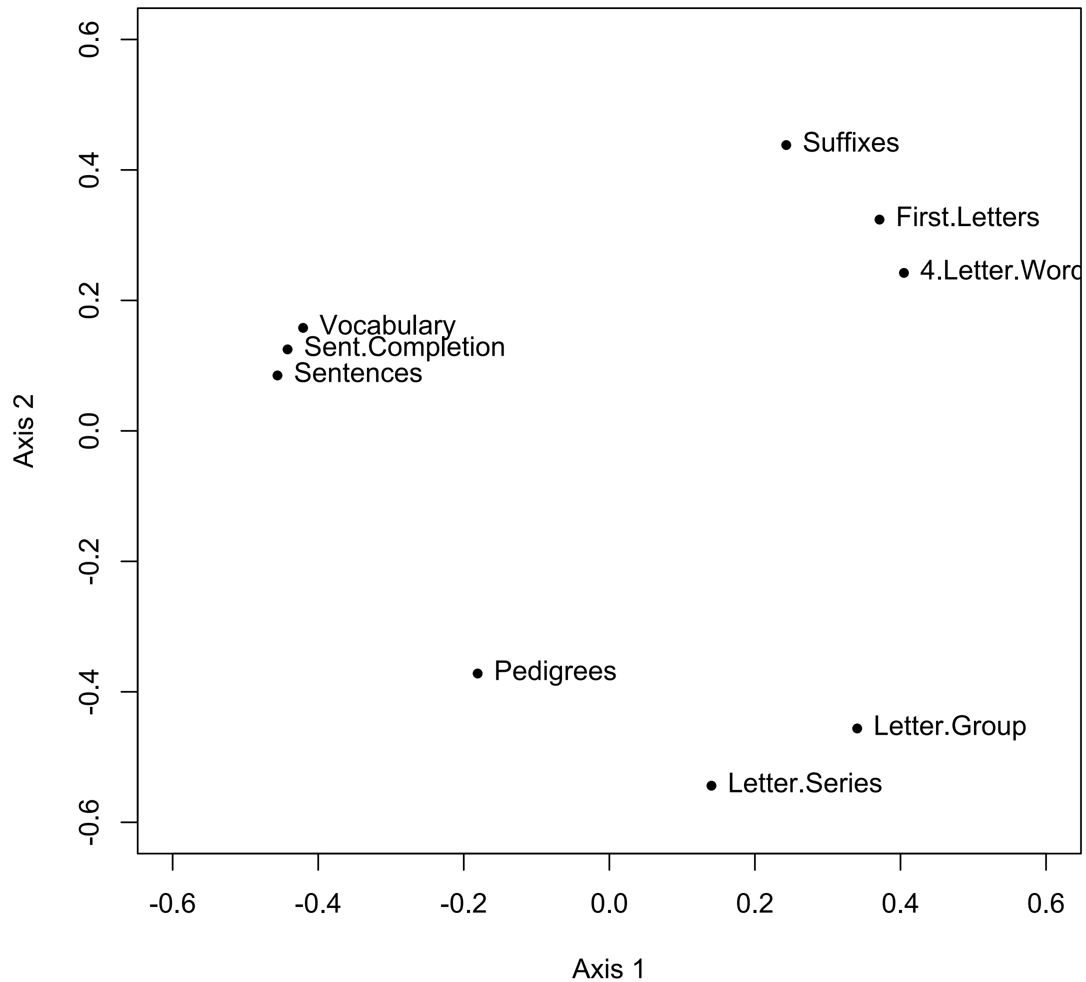
I. Convert correlations to distances

II. Multidimensional scaling will remove a general factor since it considers relative ranking of distances

# MDS

```
> Thurs.dist <- sqrt(2*(1-Thurstone))
> mdsT <- cmdscale(Thurs.dist,2)
> round(mdsT,2)
                 [,1]  [,2]
Sentences       -0.46  0.08
Vocabulary      -0.42  0.16
Sent.Completion -0.44  0.12
First.Letters    0.37  0.32
4.Letter.Words   0.40  0.24
Suffixes         0.24  0.44
Letter.Series    0.14 -0.54
Pedigrees       -0.18 -0.37
Letter.Group     0.34 -0.46
```

# MDS plot



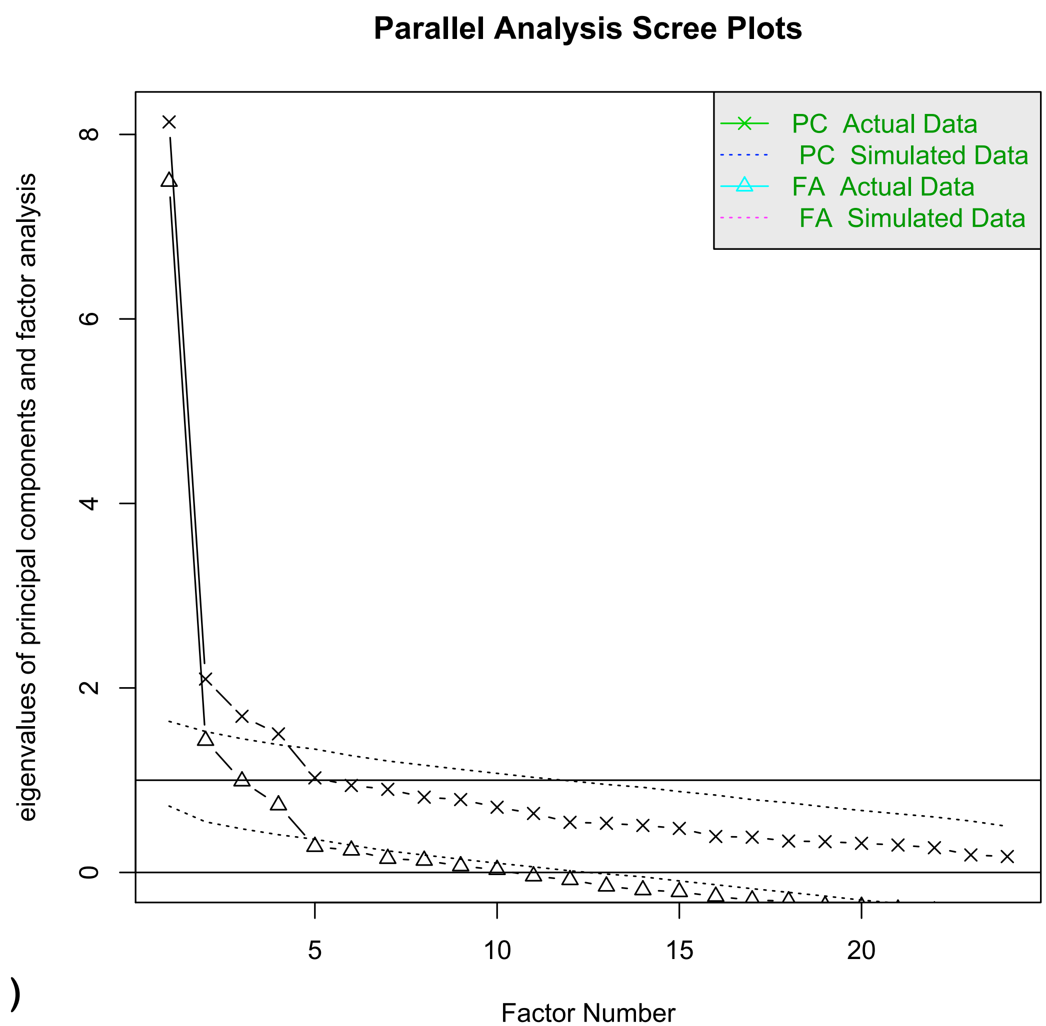

```
position <- rep(4,9)
plot(mdsT,xlim=c(-.6,.6),ylim=c(-.6,.6),ylab="Axis 2",xlab="Axis 1")
text(mdsT,rownames(mdsT),pos=position)
```

# A more complex example

I. Holzinger-Harman 24 mental ability tests

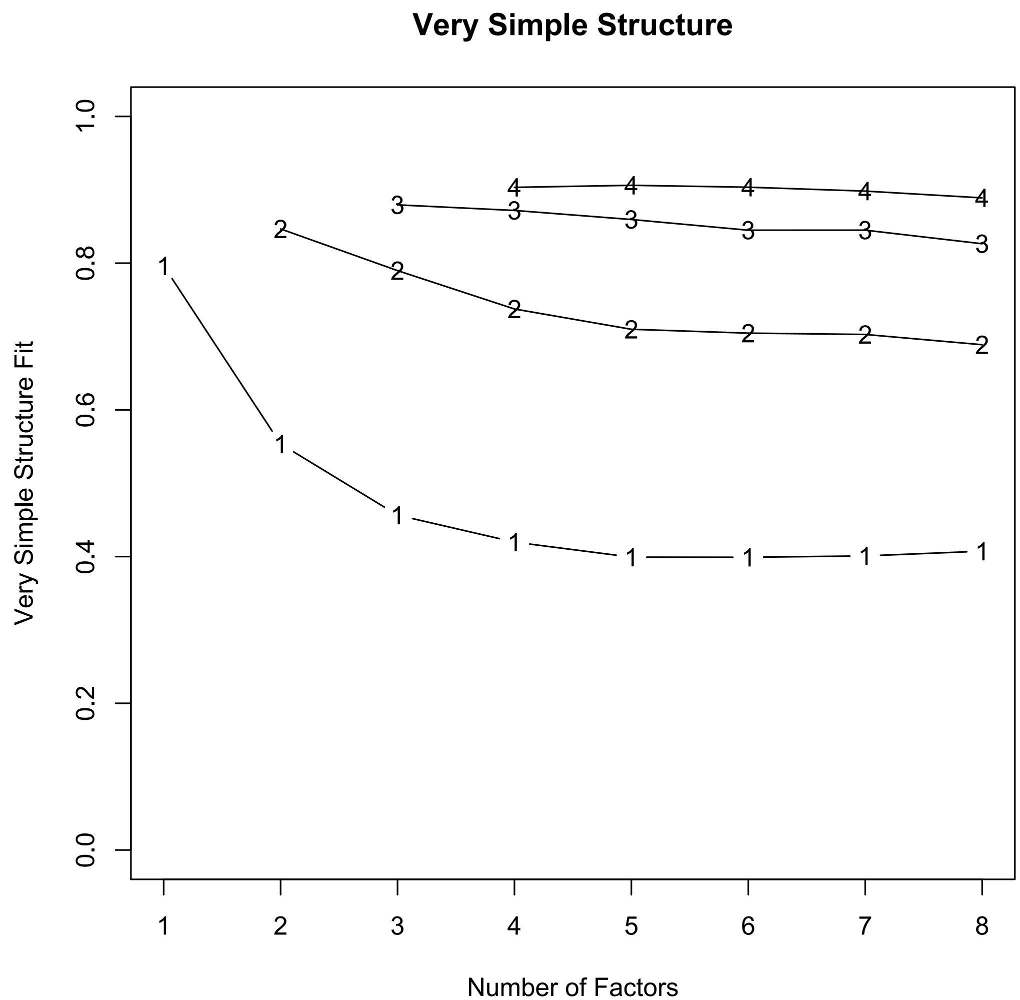
II. Compare FA, CA, MDS

# Parallel analysis




```
> hh <- Harman74.cor$cov
> fa.parallel(hh,n.obs=228)
```

# VSS says 1 big factor




```
> vss <- VSS(hh,n.obs=228)
```

# MAP suggests 4

```
> vss

Very Simple Structure
Call: VSS(x = hh, n.obs = 228)
VSS complexity 1 achieves a maximimum of 0.8  with  1
factors
VSS complexity 2 achieves a maximimum of 0.85  with  2
factors

The Velicer MAP criterion achieves a minimum of 0.03  with
4  factors

Velicer MAP
[1] 0.02 0.02 0.02 0.02 0.02 0.02 0.03 0.03

Very Simple Structure Complexity 1
[1] 0.80 0.55 0.46 0.42 0.40 0.40 0.40 0.41

Very Simple Structure Complexity 2
[1] 0.00 0.85 0.79 0.74 0.71 0.70 0.70 0.69
```

# Chi square says > 8

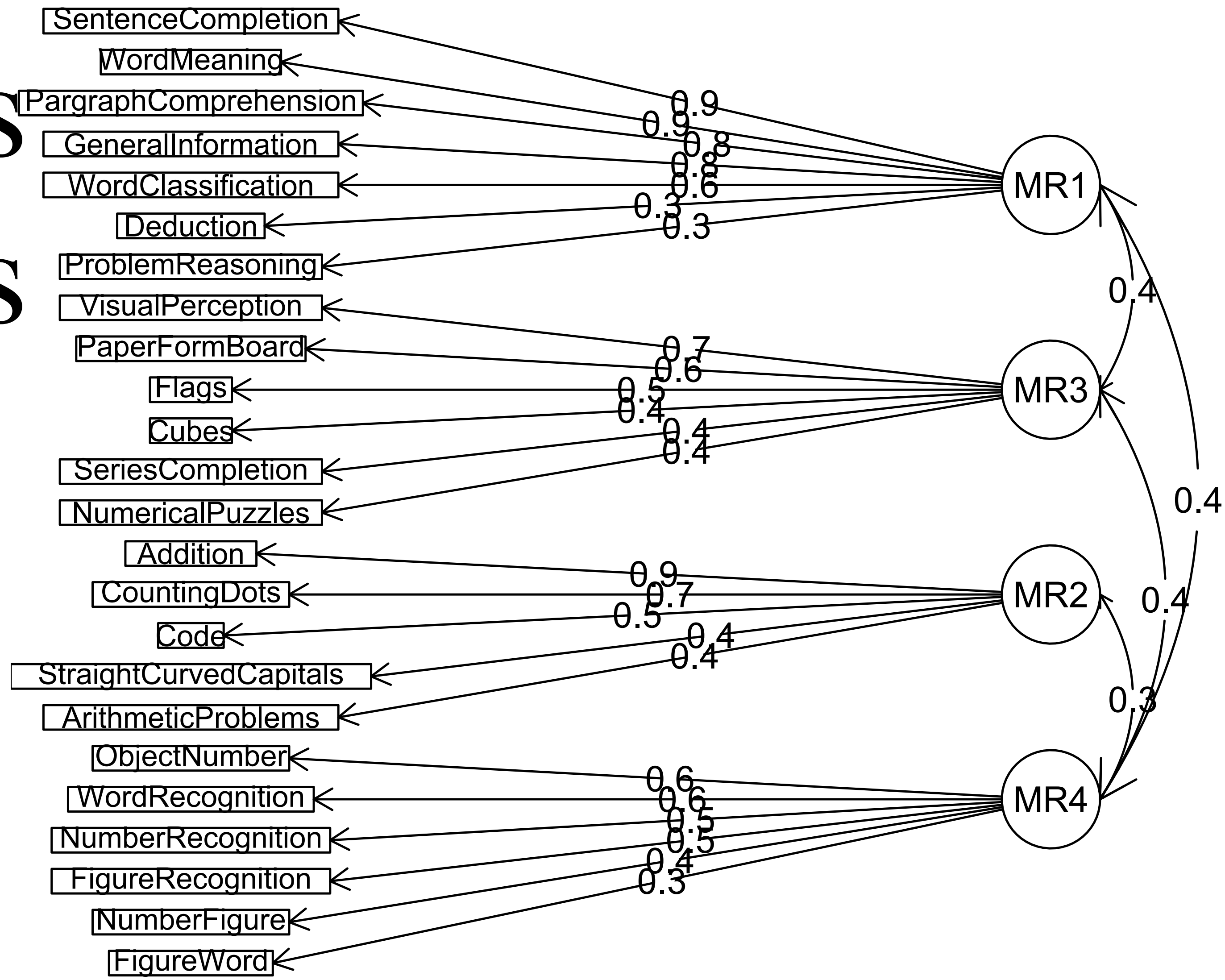
```
> vss$vss.stats[,1:3]
  dof     chisq         prob
1 252 1012.9000 9.259262e-92
2 229  693.9911 2.649458e-48
3 207  485.1213 2.232607e-24
4 186  370.8825 2.174718e-14
5 166  307.7273 1.479600e-10
6 147  266.8324 5.628353e-09
7 129  226.6602 2.396296e-07
8 112  182.6407 2.832707e-05
>
```

# 4 oblique factors

```
> fa4o <- factor.pa(hh,4,n.obs=228,rotate="oblimin")
> fa4o
                        V   PA1   PA3   PA2   PA4
VisualPerception        1        0.69
Cubes                   2        0.46
PaperFormBoard          3        0.54
Flags                   4        0.52
GeneralInformation      5  0.76
PargraphComprehension   6  0.80
SentenceCompletion      7  0.87
WordClassification      8  0.56
WordMeaning             9  0.86
Addition               10              0.86
Code                   11              0.49  0.30
CountingDots           12              0.70
StraightCurvedCapitals 13        0.42  0.47
WordRecognition        14                    0.58
NumberRecognition      15                    0.55
FigureRecognition      16        0.33        0.52
ObjectNumber           17                    0.59
NumberFigure           18                    0.43
FigureWord             19                    0.32
Deduction              20  0.33  0.31
NumericalPuzzles       21        0.37  0.33
ProblemReasoning       22  0.31  0.30
SeriesCompletion       23  0.30  0.44
ArithmeticProblems     24              0.41

 With factor correlations
     PA1  PA3  PA2  PA4
PA1 1.00 0.41 0.30 0.40
PA3 0.41 1.00 0.27 0.38
PA2 0.30 0.27 1.00 0.32
```

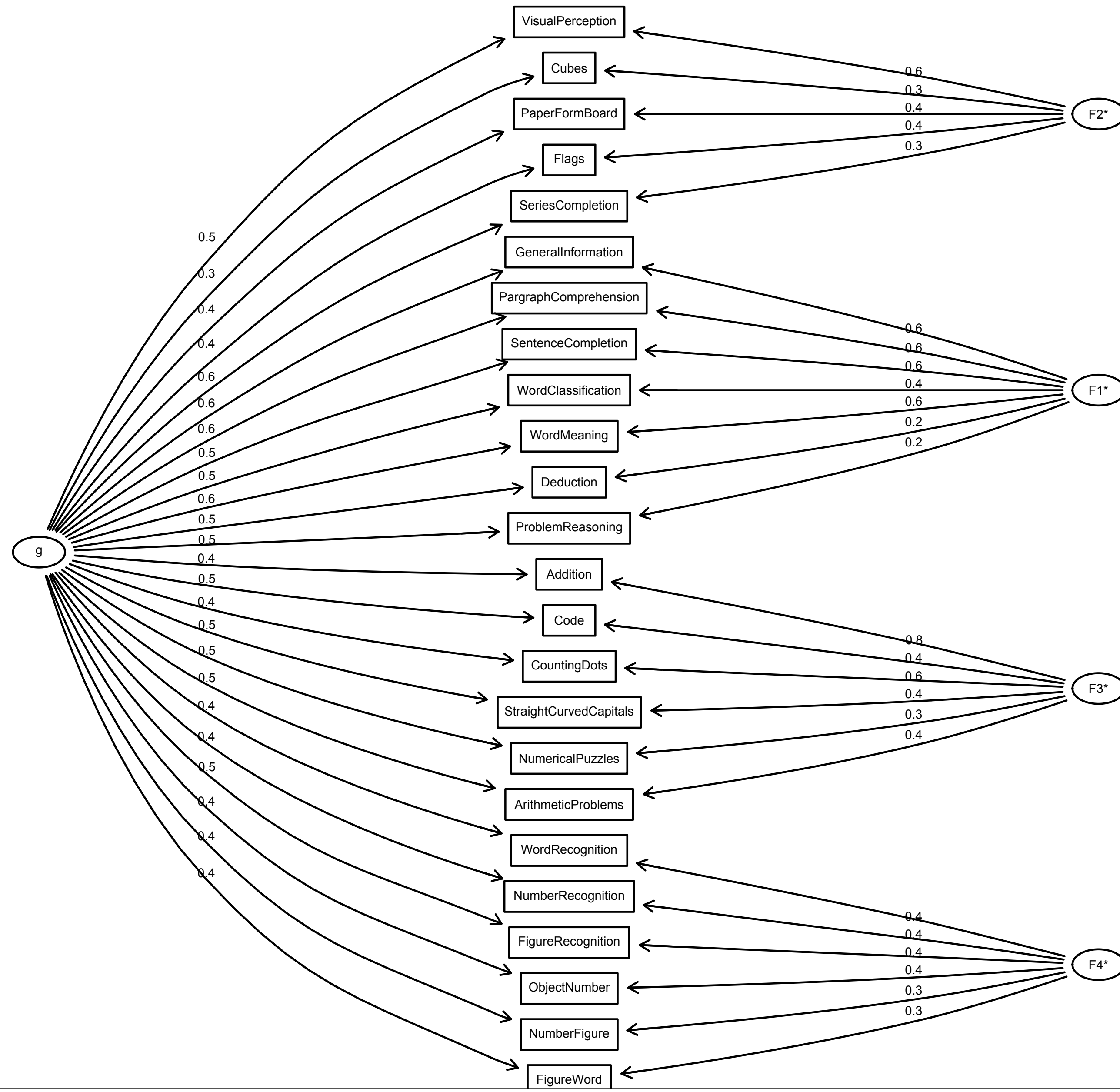
# 4 oblique factors of the Holzinger-Harman problem

4 minres factors

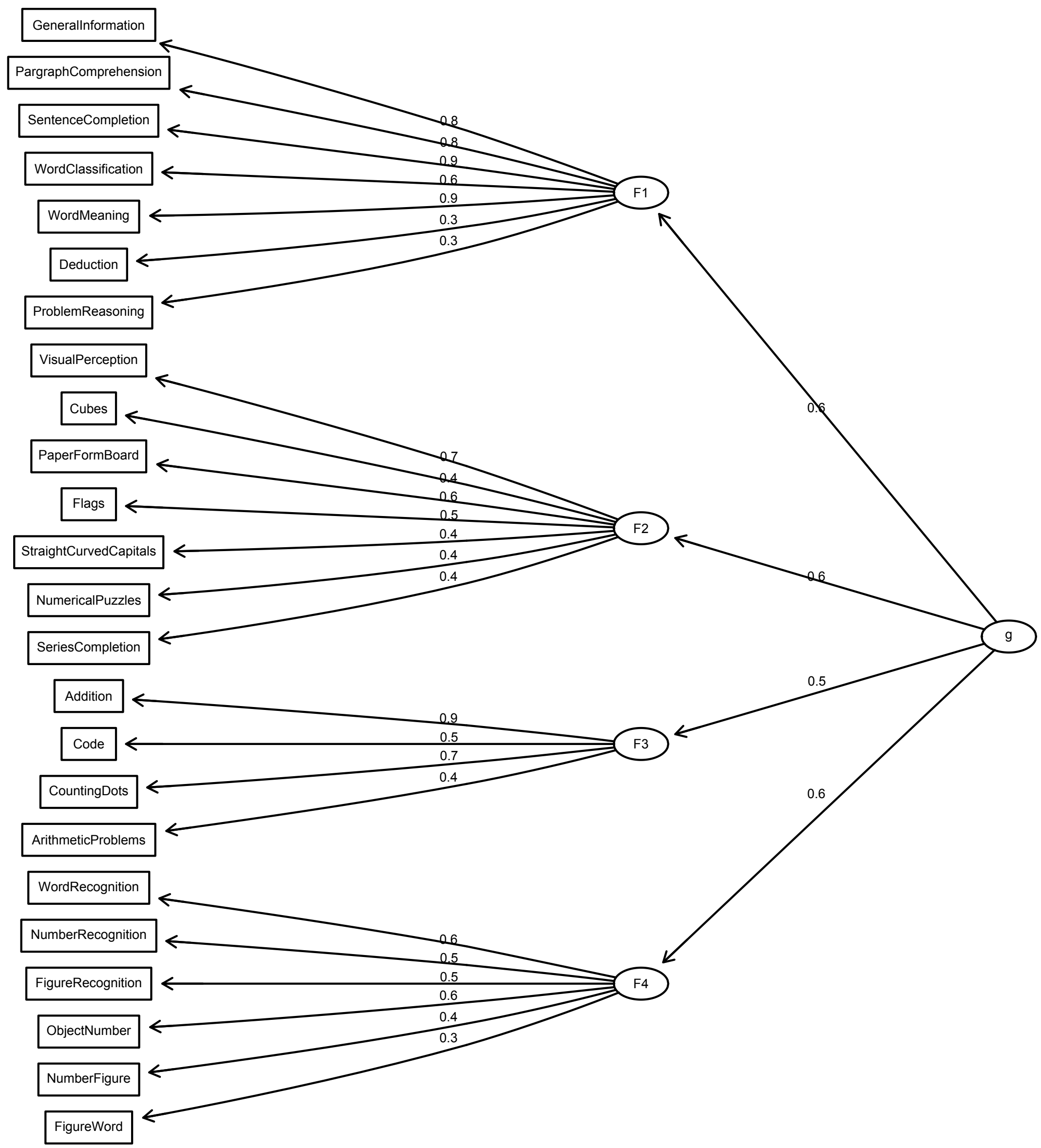


## Holzinger Harman problem -- Bifactor

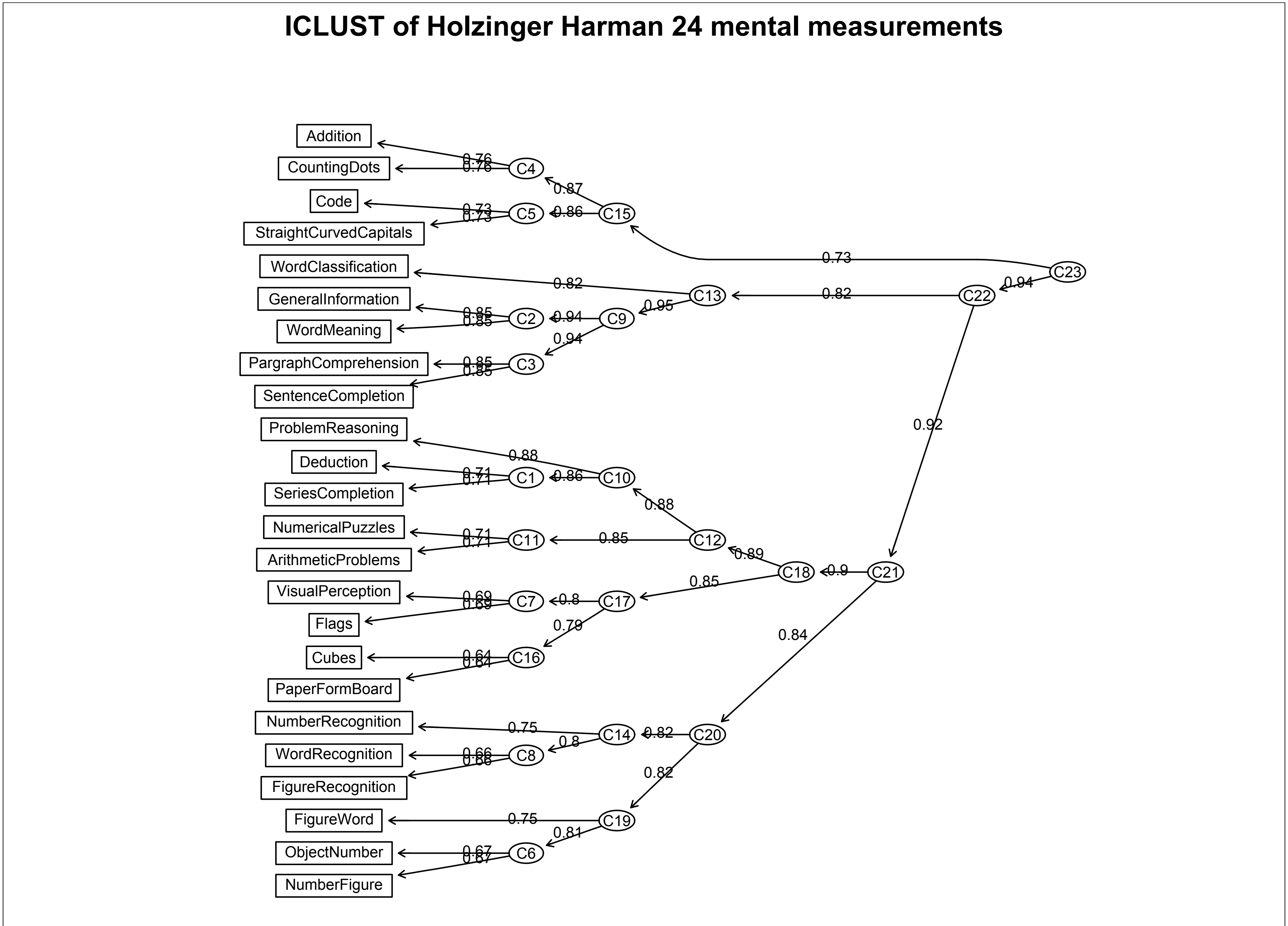
```
om4 <-omega(hh,4,title="Holzinger Harman problem -- Bifactor")
```



# Holzinger Harman problem -- Hierarchical

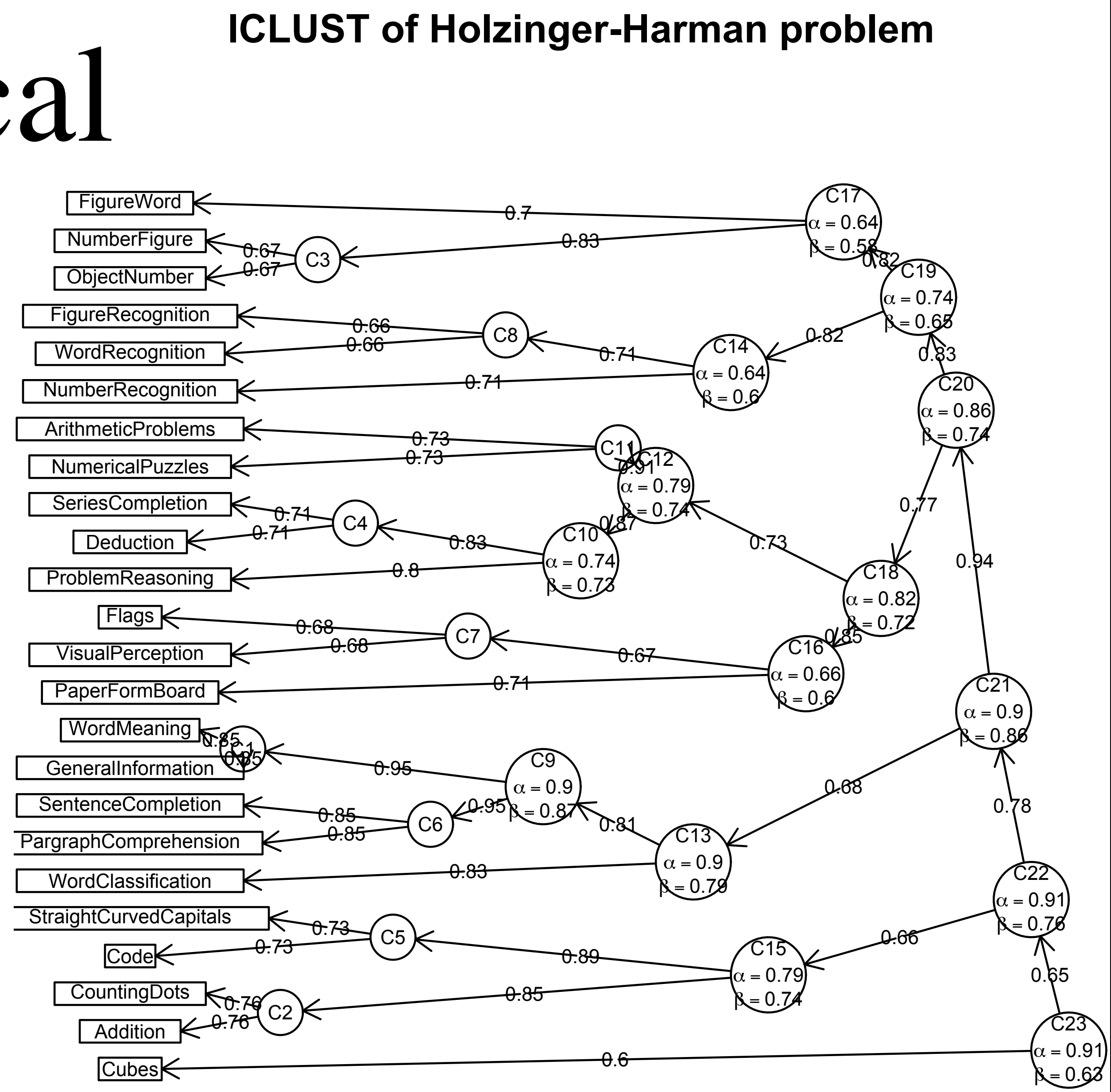



```
> om4 <-omega(hh,4,title="Holzinger Harman problem -- Hierarchical",sl=FALSE)
```

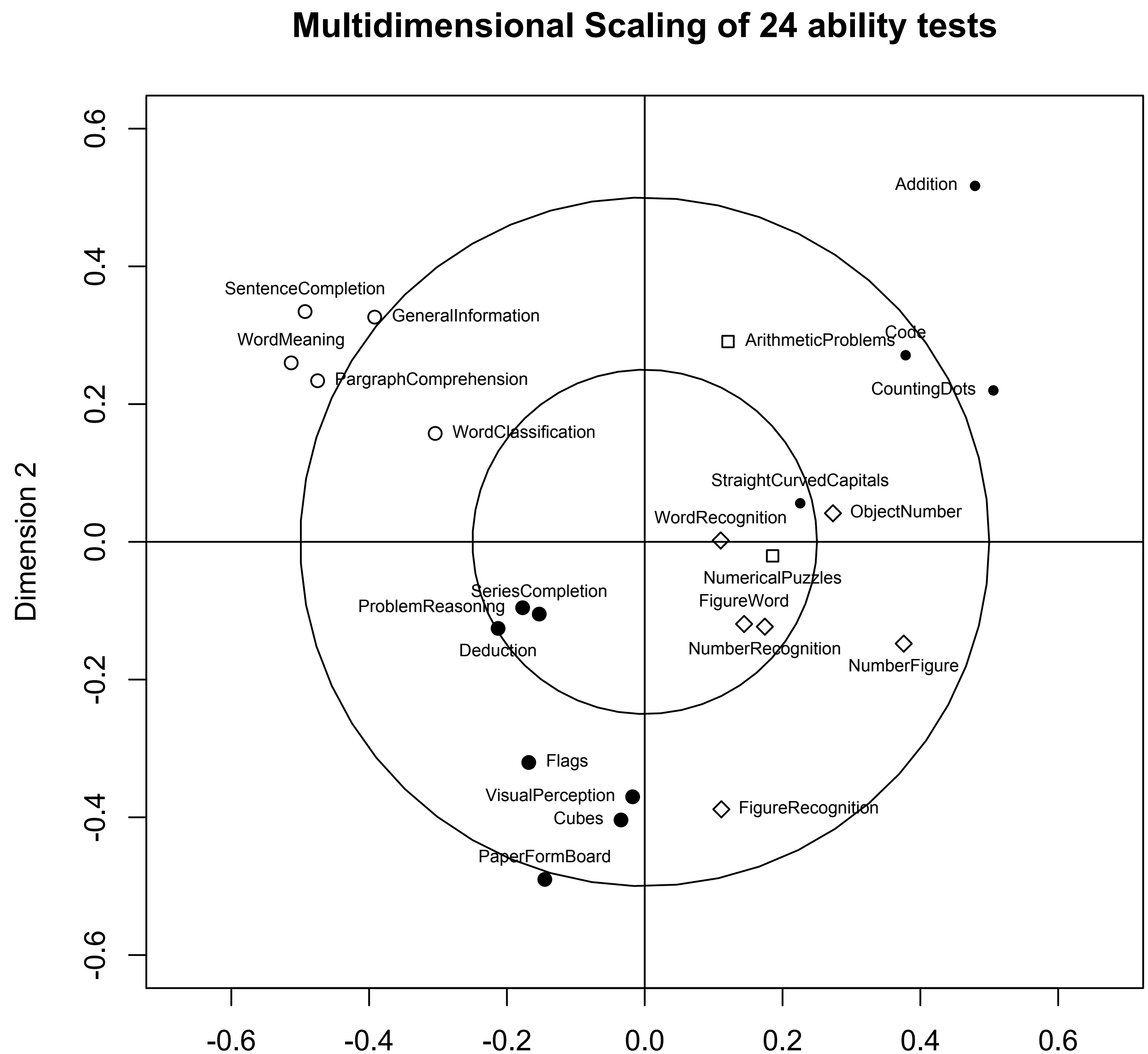



```
ic4 <- ICLUST(hh,title="ICLUST of Holzinger Harman 24 mental measurements")
```

# Hierarchical cluster analysis



# MDS of HH problem



# code for MDS plot

```
> 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)
```

# Reliability of scales and raters

I. First consider the reliability of raters using the IntraClass Correlation (see Shrout and Fleiss for the definitive discussion)

II. Types of reliability of ratings (1 or n per target)

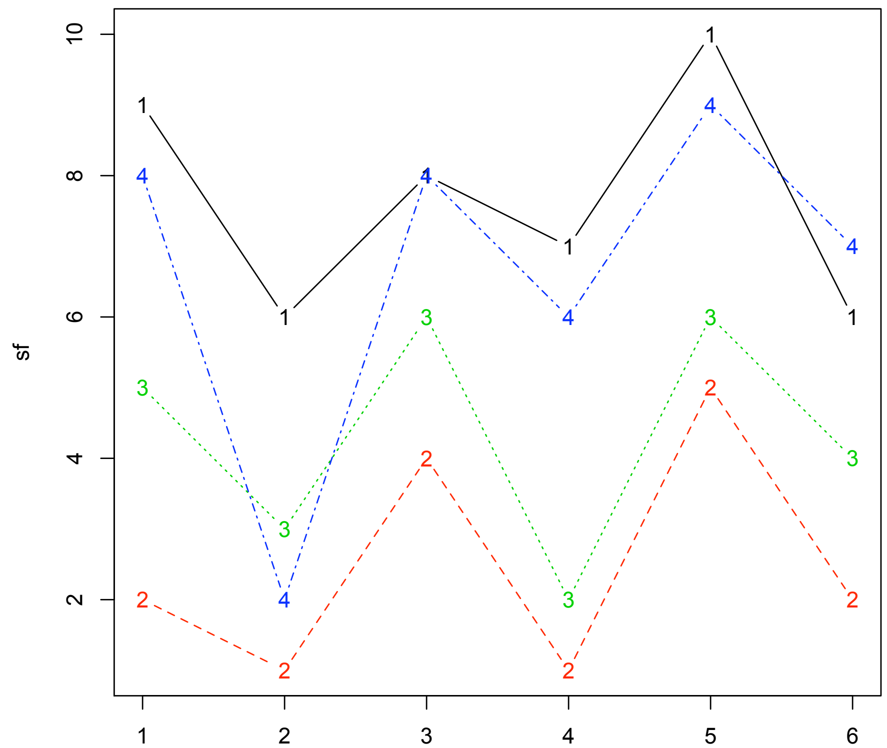
   A. each target rated by a different judge, judges are random

   B. random sample of k judges rate targets

   C. Fixed set of k judges give ratings

# 4 judges rate 6 subjects

```
> sf
   J1 J2 J3 J4
S1  9  2  5  8
S2  6  1  3  2
S3  8  4  6  8
S4  7  1  2  6
S5 10  5  6  9
S6  6  2  4  7
```



# Simple correlations (these will remove means for raters)

```
> round(cor(sf),2)
     J1   J2   J3   J4
J1 1.00 0.75 0.73 0.75
J2 0.75 1.00 0.89 0.73
J3 0.73 0.89 1.00 0.72
J4 0.75 0.73 0.72 1.00
```

# ICC

```
> ICC(sf)
                             type  ICC     F df1 df2    p lower bound upper bound
 Single_raters_absolute     ICC1 0.17  1.79   5  18 0.16       -0.13        0.72
 Single_random_raters       ICC2 0.29 11.03   5  15 0.00        0.02        0.76
 Single_fixed_raters        ICC3 0.71 11.03   5  15 0.00        0.34        0.95
 Average_raters_absolute   ICC1k 0.44  1.79   5  18 0.16       -0.88        0.91
 Average_random_raters     ICC2k 0.62 11.03   5  15 0.00        0.07        0.93
 Average_fixed_raters      ICC3k 0.91 11.03   5  15 0.00        0.68        0.99
```

# Alpha of raters

```
> alpha(sf)

Reliability analysis
Call: alpha(x = sf)

  raw_alpha std.alpha G6(smc) average_r mean  sd
       0.91      0.93    0.92      0.76   21 6.7

 Reliability if an item is dropped:
   raw_alpha std.alpha G6(smc) average_r
J1      0.88      0.91    0.89      0.78
J2      0.87      0.89    0.85      0.73
J3      0.87      0.90    0.85      0.74
J4      0.92      0.92    0.90      0.79
```

```
Item statistics
   n    r r.cor mean  sd
J1 6 0.89  0.83  7.7 1.6
J2 6 0.93  0.92  2.5 1.6
J3 6 0.92  0.91  4.3 1.6
J4 6 0.88  0.82  6.7 2.5
```

# Reliability of a single scale

```
> round(cor(bfi[,1:10],use="pairwise"),2)
      A1    A2    A3    A4    A5    C1    C2    C3    C4    C5
A1  1.00 -0.30 -0.23 -0.12 -0.19 -0.03 -0.06  0.01  0.19  0.08
A2 -0.30  1.00  0.39  0.24  0.41  0.06  0.06  0.22 -0.15 -0.12
A3 -0.23  0.39  1.00  0.27  0.45  0.07  0.12  0.16 -0.14 -0.12
A4 -0.12  0.24  0.27  1.00  0.22  0.07  0.17  0.08 -0.15 -0.17
A5 -0.19  0.41  0.45  0.22  1.00  0.10  0.06  0.20 -0.14 -0.10
C1 -0.03  0.06  0.07  0.07  0.10  1.00  0.44  0.41 -0.39 -0.23
C2 -0.06  0.06  0.12  0.17  0.06  0.44  1.00  0.35 -0.36 -0.24
C3  0.01  0.22  0.16  0.08  0.20  0.41  0.35  1.00 -0.37 -0.36
C4  0.19 -0.15 -0.14 -0.15 -0.14 -0.39 -0.36 -0.37  1.00  0.53
C5  0.08 -0.12 -0.12 -0.17 -0.10 -0.23 -0.24 -0.36  0.53  1.00
```

# Mindless reliability

```
> alpha(bfi[1:10])

Reliability analysis
Call: alpha(x = bfi[1:10])

  raw_alpha std.alpha G6(smc) average_r mean  sd
      0.19      0.25    0.44     0.032   40 4.7

 Reliability if an item is dropped:        Item statistics
   raw_alpha std.alpha G6(smc) average_r          n     r r.cor mean  sd
A1     0.290     0.354    0.51     0.057  A1 1000 0.099 -0.21  2.3 1.3
A2     0.082     0.133    0.35     0.017  A2  994 0.508  0.47  4.8 1.1
A3     0.045     0.101    0.33     0.012  A3  989 0.552  0.54  4.6 1.2
A4     0.108     0.173    0.40     0.023  A4  993 0.448  0.31  4.8 1.4
A5     0.042     0.093    0.32     0.011  A5  988 0.563  0.55  4.6 1.2
C1     0.133     0.190    0.38     0.025  C1  997 0.421  0.34  4.4 1.2
C2     0.123     0.182    0.38     0.024  C2  997 0.434  0.35  4.2 1.3
C3     0.105     0.154    0.36     0.020  C3  995 0.478  0.44  4.3 1.3
C4     0.336     0.392    0.50     0.067  C4  986 0.005 -0.24  2.6 1.4
C5     0.329     0.364    0.49     0.060  C5  997 0.077 -0.17  3.5 1.5
```

# somewhat better reliability

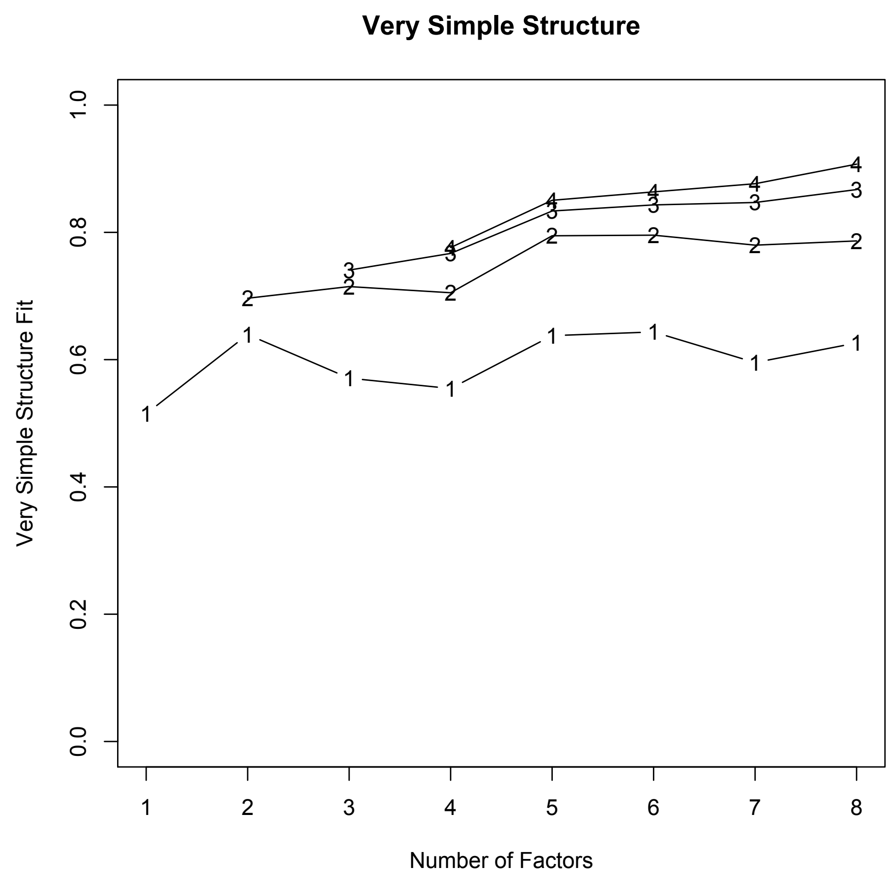
```
> keys <- make.keys(10,list(all=c(-1,2:8,-9,-10)))
> alpha(bfi[1:10],keys)

Reliability analysis
Call: alpha(x = bfi[1:10], keys = keys)

  raw_alpha std.alpha G6(smc) average_r mean  sd
      0.72      0.72    0.75      0.21   40 4.7

 Reliability if an item is dropped:
   raw_alpha std.alpha G6(smc) average_r
A1      0.72      0.72    0.74      0.23
A2      0.70      0.70    0.72      0.20
A3      0.70      0.70    0.72      0.20
A4      0.71      0.71    0.74      0.22
A5      0.70      0.70    0.72      0.21
C1      0.70      0.70    0.72      0.21
C2      0.70      0.70    0.72      0.21
C3      0.69      0.69    0.71      0.20
C4      0.67      0.68    0.70      0.19
C5      0.70      0.70    0.72      0.20

 Item statistics
      n    r r.cor mean  sd
A1 1000 0.41  0.29  2.3 1.3
A2  994 0.55  0.49  4.8 1.1
A3  989 0.55  0.49  4.6 1.2
A4  993 0.46  0.35  4.8 1.4
A5  988 0.53  0.46  4.6 1.2
C1  997 0.52  0.46  4.4 1.2
C2  997 0.54  0.47  4.2 1.3
C3  995 0.59  0.54  4.3 1.3
C4  986 0.64  0.61  2.6 1.4
C5  997 0.55  0.49  3.5 1.5
```

# Examine the items

## VSS suggests 2 factors!

## The items

`A1`
Am indifferent to the feelings of others.
`A2`
Inquire about others' well-being.
`A3`
Know how to comfort others.
`A4`
Love children.
`A5`
Make people feel at ease.
`C1`
Am exacting in my work.
`C2`
Continue until everything is perfect.
`C3`
Do things according to a plan.
`C4`
Do things in a half-way manner.
`C5`
Waste my time.



# Omega reliability

```
> om2 <- omega(bfi[1:10],2)
Warning messages:
1: In schmid(m, nfactors, pc, digits, rotate = rotate, n.obs =
n.obs,  :
  Three factors are required for identification -- general factor
loadings set to be equal. Proceed with caution.
2: In schmid(m, nfactors, pc, digits = digits, n.obs = n.obs, ...) :
  Three factors are required for identification -- general factor
loadings set to be equal. Proceed with caution.
```

```
> om2
Omega
Call: omega(m = bfi[1:10], nfactors = 2)
Alpha:  0.72
G.6:   0.75
Omega Hierarchical:   0.36
Omega Total   0.77

Schmid Leiman Factor loadings greater than  0.2
    g    F1*  F2*  h2   u2
A1- 0.21      0.30      0.87
A2  0.36      0.53 0.41 0.59
A3  0.36      0.55 0.43 0.57
A4  0.24      0.28      0.86
A5  0.36      0.54 0.42 0.58
C1  0.30 0.51      0.36 0.64
C2  0.30 0.48      0.32 0.68
C3  0.37 0.48      0.37 0.63
C4- 0.40 0.58      0.50 0.50
C5- 0.33 0.48      0.33 0.67

With eigenvalues of:
  g F1* F2*
1.1 1.3 1.1
```

# Omega h is low

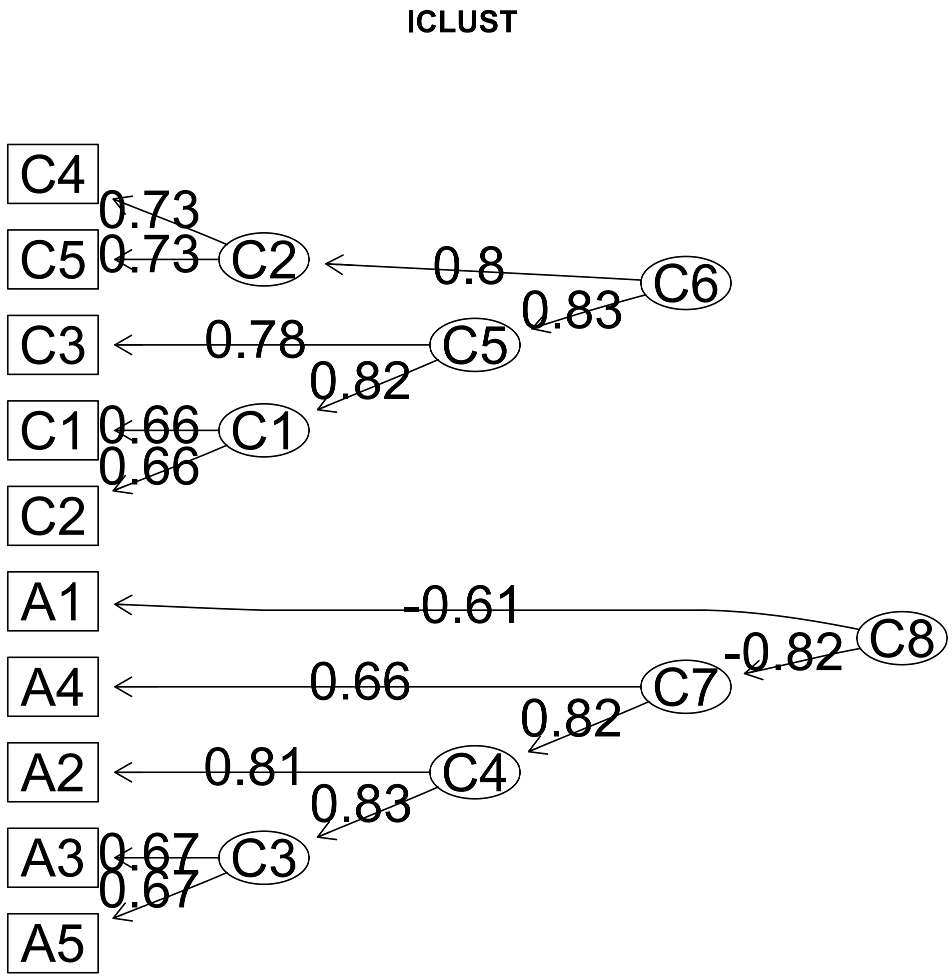
# Score 3 scales

```
> keys <- make.keys(10,list(all=c(-1,2:8,-9,-10),agree=c(-1,2:5),con=c
(6:8,-9,-10)))
> score.items(keys,bfi[1:10])
Call: score.items(keys = keys, items = bfi[1:10])

(Unstandardized) Alpha:
       all agree  con
alpha 0.72  0.65 0.74

Average item correlation:
          all agree  con
average.r 0.2  0.27 0.36

 Guttman 6* reliability:
          all agree  con
Lambda.6 0.74  0.62 0.72

Scale intercorrelations corrected for attenuation
 raw correlations below the diagonal, alpha on the diagonal
 corrected correlations above the diagonal:
       all agree  con
all   0.72  1.10 1.13
agree 0.75  0.65 0.36
con   0.83  0.25 0.74
```

# Item scale correlations

```
Item by scale correlations:
 corrected for item overlap and scale reliability
     all agree   con
A1 -0.28 -0.37 -0.12
A2  0.47  0.63  0.21
A3  0.47  0.63  0.21
A4  0.34  0.37  0.22
A5  0.44  0.58  0.20
C1  0.46  0.13  0.59
C2  0.47  0.19  0.55
C3  0.54  0.25  0.60
C4 -0.62 -0.30 -0.69
C5 -0.50 -0.23 -0.56
>
```

# ICLUST shows 2 scales



# Structural Equation modeling in R

I. sem by John Fox
II. does not do multiple group analyses
III. Mx in R is a coming attraction
IV. Using psych as a front end to sem to generate the model commands

# Using psych as front end to sem

I. Do the exploratory analysis (fa or omega) in psych

II. output includes the sem model instructions

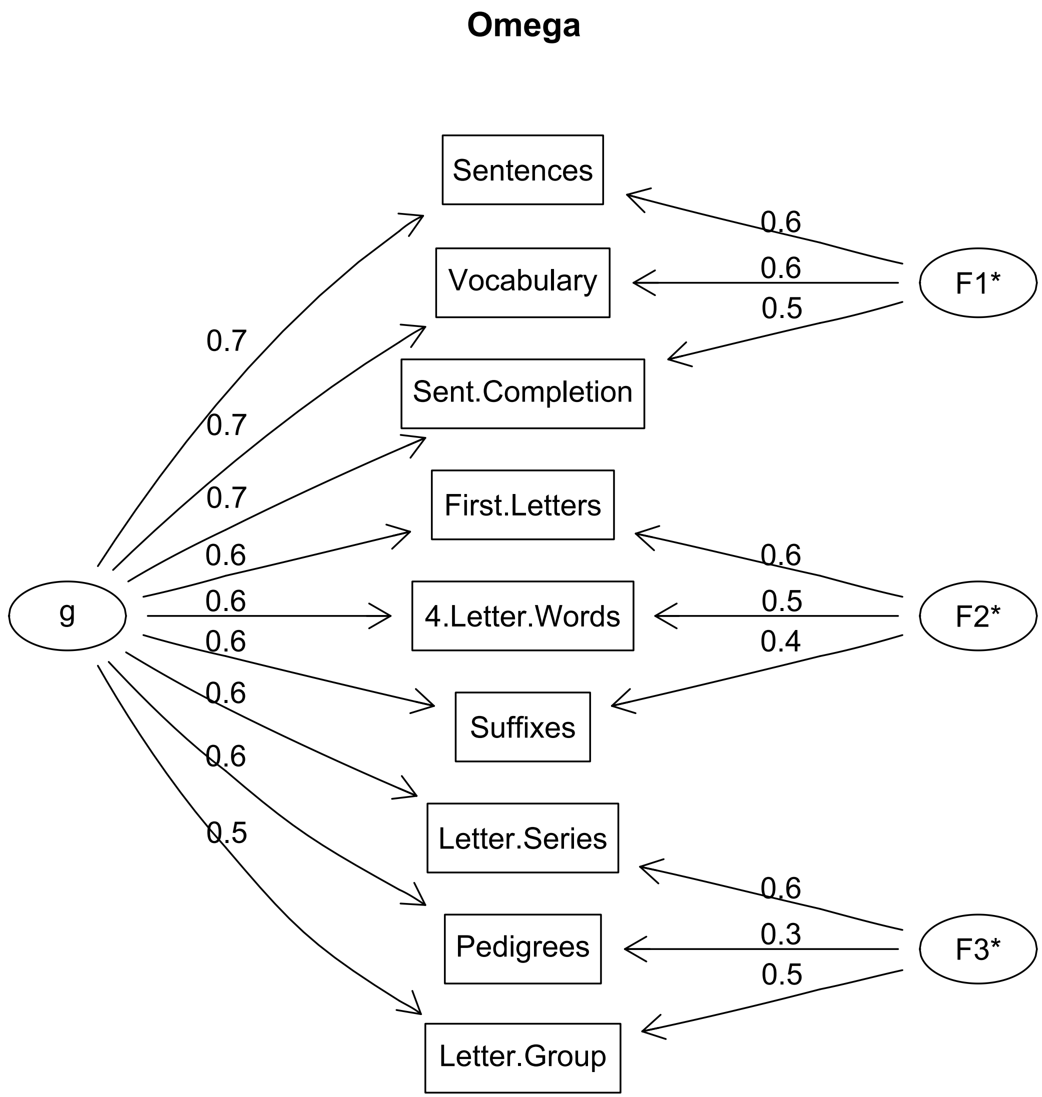
III. run sem

IV. (see the vignette on using psych for sem)

# The model

```
> om <- omega(Thurstone) #creates the path model and the model commands
> om$model
      Path                                    Parameter            Initial
Value
 [1,] "g->Sentences"                          "Sentences"          NA
 [2,] "g->Vocabulary"                         "Vocabulary"         NA
 [3,] "g->Sent.Completion"                    "Sent.Completion"    NA
 [4,] "g->First.Letters"                      "First.Letters"      NA
 [5,] "g->4.Letter.Words"                     "4.Letter.Words"     NA
 [6,] "g->Suffixes"                           "Suffixes"           NA
 [7,] "g->Letter.Series"                      "Letter.Series"      NA
 [8,] "g->Pedigrees"                          "Pedigrees"          NA
 [9,] "g->Letter.Group"                       "Letter.Group"       NA
[10,] "F1*->Sentences"                        "F1*Sentences"       NA
[11,] "F1*->Vocabulary"                       "F1*Vocabulary"      NA
[12,] "F1*->Sent.Completion"                  "F1*Sent.Completion" NA
[13,] "F2*->First.Letters"                    "F2*First.Letters"   NA
[14,] "F2*->4.Letter.Words"                   "F2*4.Letter.Words"  NA
[15,] "F2*->Suffixes"                         "F2*Suffixes"        NA
[16,] "F3*->Letter.Series"                    "F3*Letter.Series"   NA
[17,] "F3*->Pedigrees"                        "F3*Pedigrees"       NA
[18,] "F3*->Letter.Group"                     "F3*Letter.Group"    NA
[19,] "Sentences<->Sentences"                 "e1"                 NA
[20,] "Vocabulary<->Vocabulary"               "e2"                 NA
[21,] "Sent.Completion<->Sent.Completion"     "e3"                 NA
[22,] "First.Letters<->First.Letters"         "e4"                 NA
[23,] "4.Letter.Words<->4.Letter.Words"       "e5"                 NA
[24,] "Suffixes<->Suffixes"                   "e6"                 NA
[25,] "Letter.Series<->Letter.Series"         "e7"                 NA
[26,] "Pedigrees<->Pedigrees"                 "e8"                 NA
[27,] "Letter.Group<->Letter.Group"           "e9"                 NA
```

# The model


Omega
Sentences
Vocabulary
Sent.Completion
First.Letters
4.Letter.Words
Suffixes
Letter.Series
Pedigrees
Letter.Group
g
F1*
F2*
F3*
0.7
0.7
0.7
0.6
0.6
0.6
0.6
0.6
0.5
0.6
0.6
0.5
0.6
0.5
0.4
0.6
0.3
0.5

# Do the sem

```
> library(sem)
> sem.bf <- sem(om$model,Thurstone,213)
> summary(sem.bf,digits=2)

 Model Chisquare =  24   Df =  18 Pr(>Chisq) = 0.15
 Chisquare (null model) =  1102   Df =  36
 Goodness-of-fit index =  0.98
 Adjusted goodness-of-fit index =  0.94
 RMSEA index =  0.04   90% CI: (NA, 0.078)
 Bentler-Bonnett NFI =  0.98
 Tucker-Lewis NNFI =  0.99
 Bentler CFI =  1
 SRMR =  0.035
 BIC =  -72

 Normalized Residuals
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-8.2e-01 -3.3e-01 -8.9e-07  2.8e-02  1.6e-01  1.8e+00
```

# Parameter values

Parameter Estimates

| | Estimate | Std Error | z value | Pr(>\|z\|) | |
|---|---|---|---|---|---|
| Sentences | 0.77 | 0.073 | 10.57 | 0.0e+00 | Sentences <--- g |
| Vocabulary | 0.79 | 0.072 | 10.92 | 0.0e+00 | Vocabulary <--- g |
| Sent.Completion | 0.75 | 0.073 | 10.27 | 0.0e+00 | Sent.Completion <--- g |
| First.Letters | 0.61 | 0.072 | 8.43 | 0.0e+00 | First.Letters <--- g |
| 4.Letter.Words | 0.60 | 0.074 | 8.09 | 6.7e-16 | 4.Letter.Words <--- g |
| Suffixes | 0.57 | 0.071 | 8.00 | 1.3e-15 | Suffixes <--- g |
| Letter.Series | 0.57 | 0.074 | 7.63 | 2.3e-14 | Letter.Series <--- g |
| Pedigrees | 0.66 | 0.069 | 9.55 | 0.0e+00 | Pedigrees <--- g |
| Letter.Group | 0.53 | 0.079 | 6.71 | 2.0e-11 | Letter.Group <--- g |
| F1*Sentences | 0.49 | 0.085 | 5.71 | 1.1e-08 | Sentences <--- F1* |
| F1*Vocabulary | 0.45 | 0.090 | 5.00 | 5.7e-07 | Vocabulary <--- F1* |
| F1*Sent.Completion | 0.40 | 0.093 | 4.33 | 1.5e-05 | Sent.Completion <--- F1* |
| F2*First.Letters | 0.61 | 0.086 | 7.16 | 8.2e-13 | First.Letters <--- F2* |
| F2*4.Letter.Words | 0.51 | 0.085 | 5.96 | 2.5e-09 | 4.Letter.Words <--- F2* |
| F2*Suffixes | 0.39 | 0.078 | 5.04 | 4.7e-07 | Suffixes <--- F2* |
| F3*Letter.Series | 0.73 | 0.159 | 4.56 | 5.1e-06 | Letter.Series <--- F3* |
| F3*Pedigrees | 0.25 | 0.089 | 2.77 | 5.6e-03 | Pedigrees <--- F3* |
| F3*Letter.Group | 0.41 | 0.122 | 3.35 | 8.1e-04 | Letter.Group <--- F3* |
| e1 | 0.17 | 0.034 | 5.05 | 4.4e-07 | Sentences <--> Sentences |
| e2 | 0.17 | 0.030 | 5.65 | 1.6e-08 | Vocabulary <--> Vocabulary |
| e3 | 0.27 | 0.033 | 8.09 | 6.7e-16 | Sent.Completion <-->Sent.Comp |
| e4 | 0.25 | 0.079 | 3.18 | 1.5e-03 | First.Letters <-First.Letters |
| e5 | 0.39 | 0.063 | 6.13 | 8.8e-10 | 4.Letter.Words <-->4.Letter.W |
| e6 | 0.52 | 0.060 | 8.68 | 0.0e+00 | Suffixes <--> Suffixes |
| e7 | 0.15 | 0.223 | 0.67 | 5.0e-01 | Letter.Series <--> Letter.Ser |
| e8 | 0.50 | 0.060 | 8.39 | 0.0e+00 | Pedigrees <--> Pedigrees |

# Programming in R

I. Very high level language

A. interpreted at run time

B. can integrate Fortran or C++ code

II. 3 ways of developing code

A. cut and paste from an editor

B. Modify prior code by adding or changing a function

C. source from a file

D. build a package for local or global distribution

# Programming in R

I. functions and data structures

A. The output of all functions is an object that will have a certain structure

B. Part of the structure might be invisible but can be shown with the str() command.

1. f3 <- fa(Thurstone,3,rotate=oblimin)

2. str(f3)  #will show more than just asking for f3

# selected output

```
> f3 <- fa(Thurstone,3,rotate="oblimin")
> f3
Factor Analysis using method =  minres
Call: fa(r = Thurstone, nfactors = 3, rotate = "oblimin")
                item   MR1   MR2   MR3   h2   u2
Sentences          1  0.90              0.82 0.18
Vocabulary         2  0.89              0.84 0.16
Sent.Completion    3  0.84              0.74 0.26
First.Letters      4        0.85        0.73 0.27
4.Letter.Words     5        0.75        0.63 0.37
Suffixes           6        0.63        0.50 0.50
Letter.Series      7              0.84 0.72 0.28
Pedigrees          8  0.38        0.47 0.50 0.50
Letter.Group       9              0.63 0.52 0.48
                MR1  MR2  MR3
SS loadings    2.64 1.87 1.49
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 0.53
MR2 0.59 1.00 0.52
MR3 0.53 0.52 1.00
Test of the hypothesis that 3 factors are sufficient.
The degrees of freedom for the model is 12  and the objective function was  0.01
Fit based upon off diagonal values = 1
Measures of factor score adequacy
                                             [,1] [,2] [,3]
Correlation of scores with factors           0.96 0.92 0.90
Multiple R square of scores with factors     0.93 0.85 0.82
Minimum correlation of factor score estimates 0.86 0.71 0.63
```

# All the output (too much)

```
> print(f3,all=TRUE)
Factor Analysis using method =  minres
Call: fa(r = Thurstone, nfactors = 3, rotate = "oblimin")
                item   MR1   MR2   MR3   h2   u2
Sentences          1  0.90             0.82 0.18
Vocabulary         2  0.89             0.84 0.16
Sent.Completion    3  0.84             0.74 0.26
First.Letters      4        0.85       0.73 0.27
4.Letter.Words     5        0.75       0.63 0.37
Suffixes           6        0.63       0.50 0.50
Letter.Series      7              0.84 0.72 0.28
Pedigrees          8  0.38        0.47 0.50 0.50
Letter.Group       9              0.63 0.52 0.48

                MR1  MR2  MR3
SS loadings    2.64 1.87 1.49
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 0.53
MR2 0.59 1.00 0.52
MR3 0.53 0.52 1.00

Test of the hypothesis that 3 factors are sufficient.

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

Fit based upon off diagonal values = 1
Measures of factor score adequacy
                                                 [,1] [,2] [,3]
Correlation of scores with factors               0.96 0.92 0.90
Multiple R square of scores with factors         0.93 0.85 0.82
Minimum correlation of factor score estimates   0.86 0.71 0.63
$residual
                  Sentences   Vocabulary Sent.Completion First.Letters 4.Letter.Words
Sentences        0.1797970835  0.003569352     0.001056412 -0.0053521359   0.0054931042
Vocabulary       0.0035693518  0.164376546    -0.004726931  0.0015287573  -0.0013163850
Sent.Completion  0.0010564117 -0.004726931     0.264475270  0.0064257822  -0.0061643080
First.Letters   -0.0053521359  0.001528757     0.006425782  0.2708794227  -0.0007462237
4.Letter.Words   0.0054931042 -0.001316385    -0.006164308 -0.0007462237   0.3691635597
Suffixes        -0.0005413284  0.003307528    -0.006202623  0.0009046706  -0.0007702702
Letter.Series   -0.0001193089  0.005601586    -0.009592364  0.0036158583  -0.0048190635
Pedigrees       -0.0103409056 -0.003359723     0.020557454 -0.0045780810   0.0019664267
Letter.Group     0.0074095333 -0.010637633     0.007277838 -0.0037248900   0.0083593427
                   Suffixes Letter.Series    Pedigrees  Letter.Group
Sentences       -0.0005413284 -0.0001193089 -0.010340906  0.0074095333
```

```
$fit
[1] 0.9569096

$fit.off
[1] 0.9998501

$factors
[1] 3

$n.obs
[1] NA

$PVAL
[1] NA

$dof
[1] 12

$objective
[1] 0.01390055

$criteria
 objective
0.01390055         NA         NA

$Call
fa(r = Thurstone, nfactors = 3, rotate = "oblimin"

$r.scores
          [,1]      [,2]      [,3]
[1,] 1.0000000 0.6615062 0.6122765
[2,] 0.6615062 1.0000000 0.6134362
[3,] 0.6122765 0.6134362 1.0000000

$R2
[1] 0.9285797 0.8525374 0.8161127

$valid
[1] 0.9598034 0.9078568 0.8823116
```

# Using str to show what is there

```
> f3 <- fa(Thurstone,3,rotate="oblimin")
> str(f3)
List of 22
 $ residual    : num [1:9, 1:9] 0.1798 0.00357 0.00106 -0.00535 0.00549 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:9] "Sentences" "Vocabulary" "Sent.Completion" "First.Letters" ...
  .. ..$ : chr [1:9] "Sentences" "Vocabulary" "Sent.Completion" "First.Letters" ...
 $ fit         : num 0.957
 $ fit.off     : num 1
 $ factors     : num 3
 $ n.obs       : logi NA
 $ PVAL        : logi NA
 $ dof         : num 12
 $ objective   : num 0.0139
 $ criteria    : Named num [1:3] 0.0139 NA NA
  ..- attr(*, "names")= chr [1:3] "objective" "" ""
 $ Call        : language fa(r = Thurstone, nfactors = 3, rotate = "oblimin")
 $ r.scores    : num [1:3, 1:3] 1 0.662 0.612 0.662 1 ...
 $ R2          : num [1:3] 0.929 0.853 0.816
 $ valid       : num [1:3] 0.96 0.908 0.882
 $ score.cor   : num [1:3, 1:3] 1 0.571 0.574 0.571 1 ...
 $ weights     : num [1:9, 1:3] 0.35332 0.38596 0.2284 0.01086 0.00503 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:9] "Sentences" "Vocabulary" "Sent.Completion" "First.Letters" ...
  .. ..$ : NULL
 $ communality : Named num [1:9] 0.82 0.84 0.74 0.73 0.63 0.5 0.72 0.5 0.52
  ..- attr(*, "names")= chr [1:9] "Sentences" "Vocabulary" "Sent.Completion" "First.Letters" ...
 $ uniquenesses: Named num [1:9] 0.18 0.16 0.26 0.27 0.37 0.5 0.28 0.5 0.48
  ..- attr(*, "names")= chr [1:9] "Sentences" "Vocabulary" "Sent.Completion" "First.Letters" ...
 $ values      : num [1:9] 4.85 1.09 1.04 0.48 0.45 0.37 0.32 0.23 0.17
 $ loadings    : loadings [1:9, 1:3] 0.90356 0.889 0.83522 -0.00297 -0.01535 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:9] "Sentences" "Vocabulary" "Sent.Completion" "First.Letters" ...
  .. ..$ : chr [1:3] "MR1" "MR2" "MR3"
 $ fm          : chr "minres"
 $ Phi         : num [1:3, 1:3] 1 0.592 0.535 0.592 1 ...
 $ fn          : chr "fa"
 - attr(*, "class")= chr [1:2] "psych" "fa"
```

# Programming in R

I. Data types

II. operators

III. simple functions

IV. Writing functions

# Data structures

I. elements: logical, integer, real, character, factor

II. vectors: ordered sets of elements of one type

III. matrices: ordered sets of vectors (all of one type)

IV. data.frames: ordered sets of vectors, may be different types

V. lists: ordered set of anything

# Operators

I. arithmetical

1. +, -, *, /, ^, %%
2. a + b, a-b, a*b, a/b, a^b, a %%b

II. Logical

A. a==b, !a, a!=b, a>b, a<b, a>=b,a <=b

III. Matrix

A. %*% is matrix multiplication

B. %o% is outer product

# example operations

```
> a <- 2
> b <- 3
> v <- 5:10
> w <- 6:7
> v
[1]  5  6  7  8  9 10
> w
[1] 6 7
> v ^a
[1]  25  36  49  64  81 100
> w* b
[1] 18 21
> w * v
[1] 30 42 42 56 54 70
```

# Matrix operations

```
> v
[1]  5  6  7  8  9 10
> t(v)
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    5    6    7    8    9   10
> t(v)%*% v
     [,1]
[1,]  355
> v %*% t(v)
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]   25   30   35   40   45   50
[2,]   30   36   42   48   54   60
[3,]   35   42   49   56   63   70
[4,]   40   48   56   64   72   80
[5,]   45   54   63   72   81   90
[6,]   50   60   70   80   90  100
```

# Additional matrix operators

outer product

```
> v
[1]  5  6  7  8  9 10
> w
[1] 6 7
> v %o% w
     [,1] [,2]
[1,]   30   35
[2,]   36   42
[3,]   42   49
[4,]   48   56
[5,]   54   63
[6,]   60   70
```

kronecker

matrix "addition" (psych)

```
>  x <- seq(4,8,2)
> x
[1] 4 6 8

> x %+% t(x)
     [,1] [,2] [,3]
[1,]    8   10   12
[2,]   10   12   14
[3,]   12   14   16
```

# Functions

I. Operate on an object and provide a new object

II. e.g., f <- function(x) {x * 2}

# Simple functions

```
> f <- function(x) {x * 2}
> f(43)
[1] 86
> x
[1] 4 6 8
> f(x)
[1]  8 12 16
> f( v %o% w)
     [,1] [,2]
[1,]   60   70
[2,]   72   84
[3,]   84   98
[4,]   96  112
[5,]  108  126
[6,]  120  140
```

# a subset of useful functions

I. is.na(), is.null(), is.vector(), is.matrix(), is.list()

II. sum(), rowSums(), colSums(), mean(x), rowMeans(), colMeans(), max, min, median (these work on the entire matrix)

III. var, cov, cor, sd (these work on the columns of the matrix/data.frame)

IV. help.start() brings up a web page of manuals

# More useful functions

I. rep(x,n) (repeats the value x n times)

II. c(x,y) (combines x with y)

III. cbind(x,y) combines column wise

IV. rbind(x,y) combines rowwise

V. seq(a,b,c) sequence from a to b stepping by c

# sums on matrices and data.frames

```
> z <- f( v %o% w)
> z
     [,1] [,2]
[1,]   60   70
[2,]   72   84
[3,]   84   98
[4,]   96  112
[5,]  108  126
[6,]  120  140
```

```
> sum(z)
[1] 1170
> min(z)
[1] 60
> max(z)
[1] 140
> median(z)
[1] 97
```

```
> rowSums(z)
[1] 130 156 182 208 234 260
> colSums(z)
[1] 540 630
> mean(z)
[1] 97.5
> rowMeans(z)
[1]  65  78  91 104 117 130
```

# Basic stats functions, part 2

```
> var(z)
     [,1] [,2]
[1,]  504  588
[2,]  588  686
> cov(z)
     [,1] [,2]
[1,]  504  588
[2,]  588  686
> cor(z)
     [,1] [,2]
[1,]    1    1
[2,]    1    1
> sd(z)
[1] 22.44994 26.19160
> z
     [,1] [,2]
[1,]   60   70
[2,]   72   84
[3,]   84   98
[4,]   96  112
[5,]  108  126
[6,]  120  140
```

# ?cor

```
 var(x, y = NULL, na.rm = FALSE, use)

cov(x, y = NULL, use = "everything",
    method = c("pearson", "kendall",
"spearman"))

cor(x, y = NULL, use = "everything",
     method = c("pearson", "kendall",
"spearman"))

cov2cor(V)
```

# More on cor

```
x
a numeric vector, matrix or data frame.
y
NULL (default) or a vector, matrix or data frame with
compatible dimensions to x. The default is equivalent to y = x
(but more efficient).
na.rm
logical. Should missing values be removed?
use
an optional character string giving a method for computing
covariances in the presence of missing values. This must be (an
abbreviation of) one of the strings "everything", "all.obs",
"complete.obs", "na.or.complete", or "pairwise.complete.obs".
method
a character string indicating which correlation coefficient (or
covariance) is to be computed. One of "pearson" (default),
"kendall", or "spearman", can be abbreviated.
V
symmetric numeric matrix, usually positive definite such as a
covariance matrix.
```

# row and col as functions

```
> r <- .8
> R <- diag(1,8)
> R
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
[1,]    1    0    0    0    0    0    0    0
[2,]    0    1    0    0    0    0    0    0
[3,]    0    0    1    0    0    0    0    0
[4,]    0    0    0    1    0    0    0    0
[5,]    0    0    0    0    1    0    0    0
[6,]    0    0    0    0    0    1    0    0
[7,]    0    0    0    0    0    0    1    0
[8,]    0    0    0    0    0    0    0    1
> R <- r^(abs(row(R)-col(R)))
> round(R,2)
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
[1,] 1.00 0.80 0.64 0.51 0.41 0.33 0.26 0.21
[2,] 0.80 1.00 0.80 0.64 0.51 0.41 0.33 0.26
[3,] 0.64 0.80 1.00 0.80 0.64 0.51 0.41 0.33
[4,] 0.51 0.64 0.80 1.00 0.80 0.64 0.51 0.41
[5,] 0.41 0.51 0.64 0.80 1.00 0.80 0.64 0.51
[6,] 0.33 0.41 0.51 0.64 0.80 1.00 0.80 0.64
[7,] 0.26 0.33 0.41 0.51 0.64 0.80 1.00 0.80
[8,] 0.21 0.26 0.33 0.41 0.51 0.64 0.80 1.00
```

# Yet more stats functions

I. sample(n, N, replace=TRUE)

II. eigen(X) (eigen value decomposition of X)

III. solve(X) (inverse of X)

IV. solve (X,Y) Regression of Y on X

# Creating a matrix

```
> x <- matrix(sample(10,50,replace=TRUE),ncol=5)
> x
      [,1] [,2] [,3] [,4] [,5]
 [1,]   10    3    4    4    6
 [2,]    3   10    8    8    9
 [3,]    1    6    5    8    5
 [4,]    9    1    3    5    3
 [5,]    6    8    3    5    1
 [6,]    8    6   10    1   10
 [7,]   10    5   10    2    1
 [8,]    9    3    2    2    9
 [9,]    6   10    2    9    4
[10,]    1    8    8    2    6
```

## standardize it

```
> z <- scale(x)
> z
             [,1]       [,2]       [,3]       [,4]       [,5]
 [1,]  1.04838349 -0.9819805 -0.4678877 -0.2059329  0.1852621
 [2,] -0.93504473  1.3093073  0.7798129  1.1669533  1.1115724
 [3,] -1.50173851  0.0000000 -0.1559626  1.1669533 -0.1235080
 ...
 [9,] -0.08500407  1.3093073 -1.0917380  1.5101749 -0.4322782
[10,] -1.50173851  0.6546537  0.7798129 -0.8923761  0.1852621
attr(,"scaled:center")
[1] 6.3 6.0 5.5 4.6 5.4
attr(,"scaled:scale")
[1] 3.529243 3.055050 3.205897 2.913570 3.238655
```

# Just center it

```
> c <- scale(x,scale=FALSE)

> c
      [,1] [,2] [,3] [,4] [,5]
 [1,]  3.7   -3 -1.5 -0.6  0.6
 [2,] -3.3    4  2.5  3.4  3.6
 [3,] -5.3    0 -0.5  3.4 -0.4
 [4,]  2.7   -5 -2.5  0.4 -2.4
 [5,] -0.3    2 -2.5  0.4 -4.4
 [6,]  1.7    0  4.5 -3.6  4.6
 [7,]  3.7   -1  4.5 -2.6 -4.4
 [8,]  2.7   -3 -3.5 -2.6  3.6
 [9,] -0.3    4 -3.5  4.4 -1.4
[10,] -5.3    2  2.5 -2.6  0.6
attr(,"scaled:center")
[1] 6.3 6.0 5.5 4.6 5.4
```

# Find the covariance and inverse

```
>  c <- cov(x)
> round(c,2)
      [,1]  [,2]  [,3]  [,4]  [,5]
[1,] 12.46 -6.89 -1.61 -4.53 -1.58
[2,] -6.89  9.33  2.11  4.11  0.56
[3,] -1.61  2.11 10.28 -3.89  2.22
[4,] -4.53  4.11 -3.89  8.49 -1.60
[5,] -1.58  0.56  2.22 -1.60 10.49
```

```
> round(solve(c),2)
     [,1]  [,2]  [,3]  [,4]  [,5]
[1,] 0.15  0.08  0.02  0.06  0.02
[2,] 0.08  0.23 -0.07 -0.10  0.00
[3,] 0.02 -0.07  0.16  0.12 -0.01
[4,] 0.06 -0.10  0.12  0.26  0.03
[5,] 0.02  0.00 -0.01  0.03  0.10
```

# Flow control

I. if(condition) {then do this} else {do this}

II. for (condition) do {expression}

  A. for (i in 1:n} do {x <- x + 1}

III. while (condition) {expression}

# conditionals

I. (a & b) vs. (a && b)

II. (a | b) vs. (a || b)

```
a <- 1
> if (a & b) {print ("hello")} else {print("goodby")}
Error: object 'b' not found
> if (a && b ) {print ("hello")} else {print("goodby")}
[1] "goodby"
> if (a | b) {print ("hello")} else {print("goodby")}
Error: object 'b' not found

> if (a || b) {print ("hello")} else {print("goodby")}
Error: object 'b' not found
> a <- 1
> if (a || b) {print ("hello")} else {print("goodby")}
[1] "hello"
>
```

# simple control

```
> a <- 1
> b <- 2
> c <- 3
> k <- 10
> x <-1
> if(x == a) {print("x is the same as a and has
value",x)} else {print ("x is not equal to a")}

> x <- 3
> if(x == a) {print("x is the same as a and has
value",x)} else {print ("x is not equal to a")}
[1] "x is not equal to a"
>
```

# Make that a function

```
> f1 <- function(x,y) {if(x == y) {print
("x is the same as y and has value",x)}
else {print ("x is not equal to y")}}
> f1(3,4)
[1] "x is not equal to y"
> f1(5,5)
[1] "x is the same as y and has value"
```

# Simple functions:part 2

I. Find the squared multiple correlation of a variable with all the other variables in a matrix.

II. The $R^2$ is 1- the residual variance

# The essence of the function

```
SMC <- function(R) {
R.inv <- solve(R)
SMC <- 1 - 1/diag(R.inv)}

> S <-cor(attitude)
> SMC(S)      #does not show anything

> round(SMC(S),2) #but this does
    rating complaints privileges   learning     raises 
critical    advance 
      0.73       0.77       0.38       0.62       0.68 
0.19       0.52
```

# Add a return

```
SMC <- function(R) {
R.inv <- solve(R)
SMC <- 1 - 1/diag(R.inv)
return(SMC)}

> SMC(S)
    rating complaints privileges   learning     raises
critical    advance
 0.7326020  0.7700868  0.3831176  0.6194561  0.6770498
0.1881465  0.5186447
```

# Allow it to find R

```
SMC <- function(R) {
if(dim(R)[1] !=dim(R)[2]) {R <-cor(R)}
R.inv <- solve(R)
SMC <- 1 - 1/diag(R.inv)
return(SMC)}
```

```
> SMC(attitude)
    rating complaints privileges   learning     raises
critical    advance
 0.7326020  0.7700868  0.3831176  0.6194561  0.6770498
0.1881465  0.5186447
```

# Clean up the output

```
SMC <- function(R,digits=2) {
if(dim(R)[1] !=dim(R)[2]) {R <-cor(R)}
R.inv <- solve(R)
SMC <- 1 - 1/diag(R.inv)
return(round(SMC,digits))}

> SMC(attitude)
    rating complaints privileges   learning     raises   critical
advance
      0.73       0.77       0.38       0.62       0.68       0.19
0.52
>
```

# Check for poor input

```
>  att <- data.frame(attitude[1:3],attitude[1:3])
> SMC(att)
Error in solve.default(R) : 
  Lapack routine dgesv: system is exactly singular
```

# Add checks for weird matrices

```
SMC <- function(R,digits=2) {
 p <- dim(R)[2]
 if (dim(R)[1] != p) {R <-cor(R)}
R.inv <- try(solve(R),TRUE)
if(class(R.inv)== as.character("try-error")) {SMC <- rep(1,p)
 warning("Correlation matrix not invertible, smc's returned as 1s")}
else  {smc <- 1 -1/diag(R.inv)
SMC <- 1 - 1/diag(R.inv)}
return(round(SMC,digits))}

> SMC(att)
[1] 1 1 1 1 1 1
Warning message:
In SMC(att) : Correlation matrix not invertible, smc's returned as
1s

> SMC(attitude)
    rating complaints privileges   learning     raises   critical
      0.73       0.77       0.38       0.62       0.68       0.19
```

# Further checks

## Input is a covariance matrix

```
> SMC(cov(attitude))
    rating complaints privileges   learning     raises   critical 
advance 
    -38.62     -39.76     -91.35     -51.42     -33.91     -78.49 
-49.96 
```

## Input is raw data or correlations

```
> SMC(cor(attitude))
    rating complaints privileges   learning     raises   critical 
advance 
      0.73       0.77       0.38       0.62       0.68       0.19 
0.52 
> SMC(attitude)
    rating complaints privileges   learning     raises   critical 
advance 
      0.73       0.77       0.38       0.62       0.68       0.19 
0.52 
```

# Final version

```
#modified Dec 10, 2008 to return 1 on diagonal if non-invertible
#modifed March 20, 2009 to return smcs * variance if covariance matrix
is desired
#modified April 8, 2009 to remove bug introduced March 10 when using
covar from data
"smc" <-
function(R,covar =FALSE) {
failed=FALSE
 p <- dim(R)[2]
 if (dim(R)[1] != p) {if(covar) {C <- cov(R, use="pairwise")
                                 vari <- diag(C)
                                 R <- cov2cor(C)
                                 } else {R <- cor(R,use="pairwise")}}
else {vari <- diag(R)
                                 R <- cov2cor(R)
      if (!is.matrix(R)) R <- as.matrix(R)}
 R.inv <- try(solve(R),TRUE)
 if(class(R.inv)== as.character("try-error")) {smc <- rep(1,p)
 warning("Correlation matrix not invertible, smc's returned as 1s")}
else  {smc <- 1 -1/diag(R.inv)
 if(covar) {smc <- smc * vari}}
 return(smc)
 }
```

# Creating a new function

I. Is there a base function to modify?

II. Consider the case of modifying Promax rotation to allow for any target matrix

III. original promax (inside the factanal package) had been modified to report the factor correlation.

IV. This version was created with the assistance of Pat Shrout and Steve Miller

# promax

```
> promax
function (x, m = 4)
{
    if (ncol(x) < 2)
        return(x)
    dn <- dimnames(x)
    xx <- varimax(x)
    x <- xx$loadings
    Q <- x * abs(x)^(m - 1)
    U <- lm.fit(x, Q)$coefficients
    d <- diag(solve(t(U) %*% U))
    U <- U %*% diag(sqrt(d))
    dimnames(U) <- NULL
    z <- x %*% U
    U <- xx$rotmat %*% U
    dimnames(z) <- dn
    class(z) <- "loadings"
    list(loadings = z, rotmat = U)
}
<environment: namespace:stats>
```

# Promax

```
> Promax
function (x, m = 4)
{
    if (!is.matrix(x) & !is.data.frame(x)) {
        if (!is.null(x$loadings))
            x <- as.matrix(x$loadings)
    }
    else {
        x <- x
    }
    if (ncol(x) < 2)
        return(x)
    dn <- dimnames(x)
    xx <- varimax(x)
    x <- xx$loadings
    Q <- x * abs(x)^(m - 1)
    U <- lm.fit(x, Q)$coefficients
    d <- diag(solve(t(U) %*% U))
    U <- U %*% diag(sqrt(d))
    dimnames(U) <- NULL
    z <- x %*% U
    U <- xx$rotmat %*% U
    ui <- solve(U)
    Phi <- ui %*% t(ui)
    dimnames(z) <- dn
    class(z) <- "loadings"
    result <- list(loadings = z, rotmat = U, Phi = Phi)
    class(result) <- c("psych", "fa")
    return(result)}
```

# target.rot

```
"target.rot" <-
function (x, keys=NULL,m = 4)
{
 if(!is.matrix(x) & !is.data.frame(x) )  {
        if(!is.null(x$loadings)) x <- as.matrix(x$loadings)
      } else {x <- x}
    if (ncol(x) < 2)
        return(x)
    dn <- dimnames(x)
    if(is.null(keys)) {xx <- varimax(x)
    x <- xx$loadings
    Q <- x * abs(x)^(m - 1)} else {Q <- keys}
    U <- lm.fit(x, Q)$coefficients
    d <- diag(solve(t(U) %*% U))
    U <- U %*% diag(sqrt(d))
    dimnames(U) <- NULL
    z <- x %*% U
    if (is.null(keys)) {U <- xx$rotmat %*% U } else {U <- U}
    ui <- solve(U)
    Phi <- ui %*% t(ui)
    dimnames(z) <- dn
    class(z) <- "loadings"
    result <- list(loadings = z, rotmat = U,Phi = Phi)
    class(result) <- c("psych","fa")
    return(result)
}
```

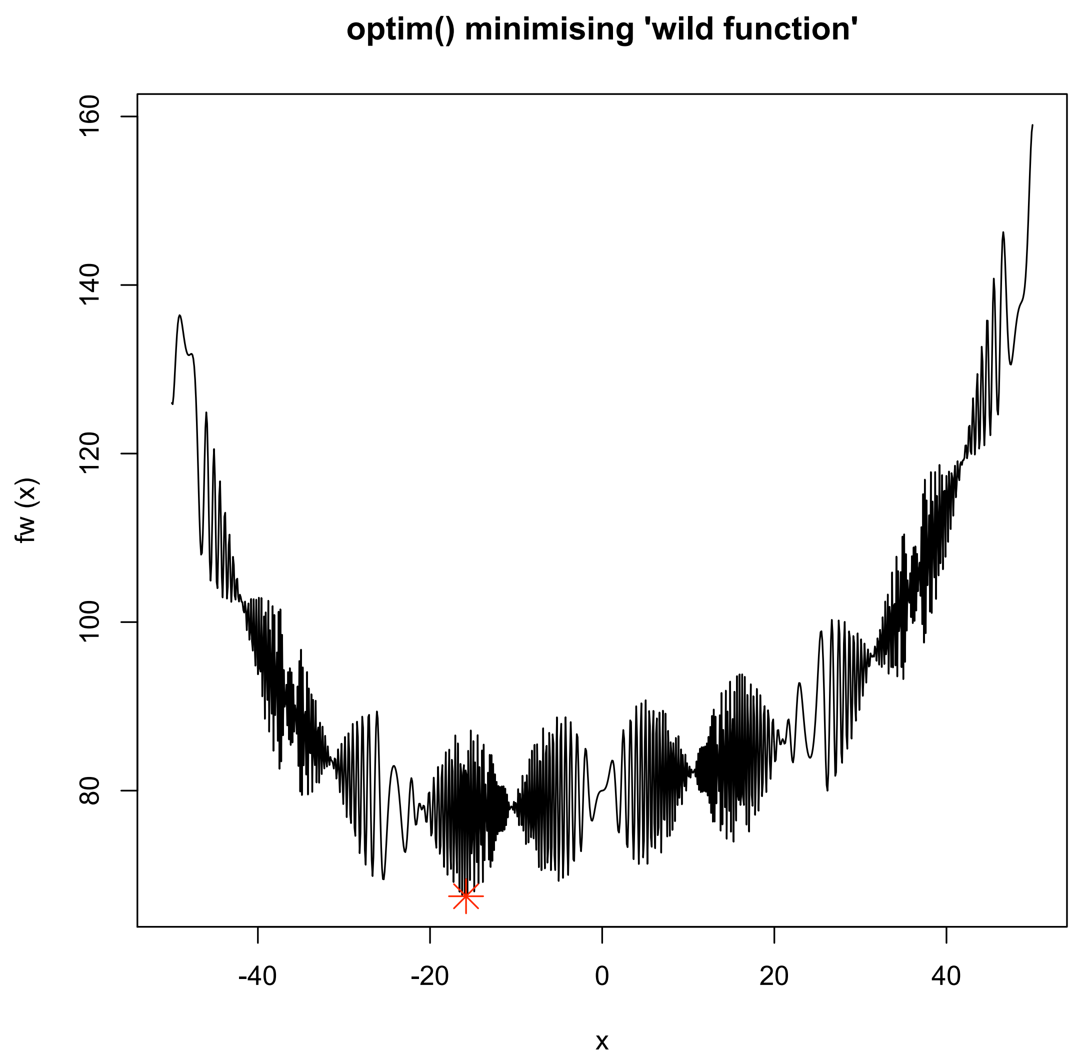
# optim as “solver” for R

I. Many statistical functions are not closed form but rather are solved iteratively.

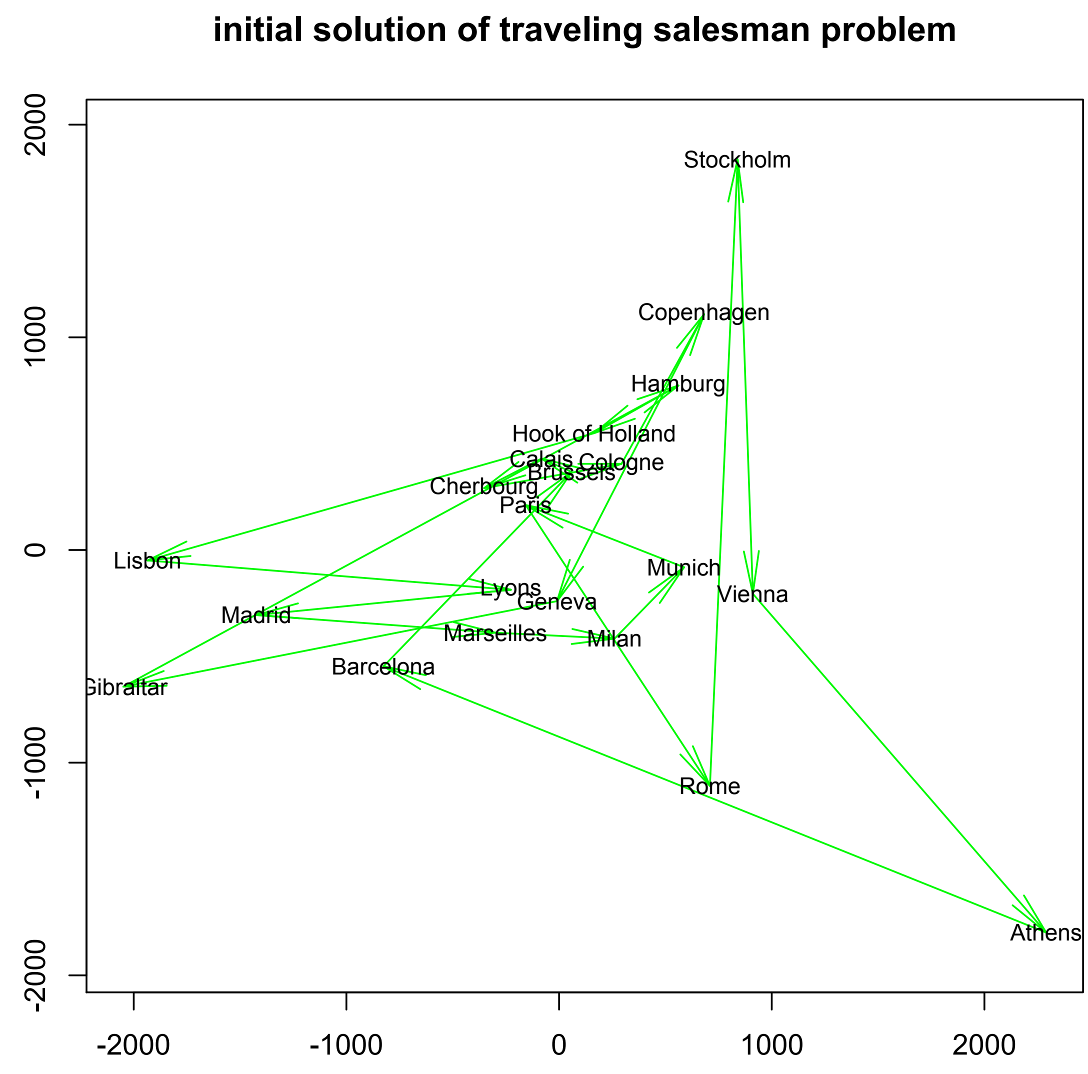
II. We start with a good guess and then minimize the function

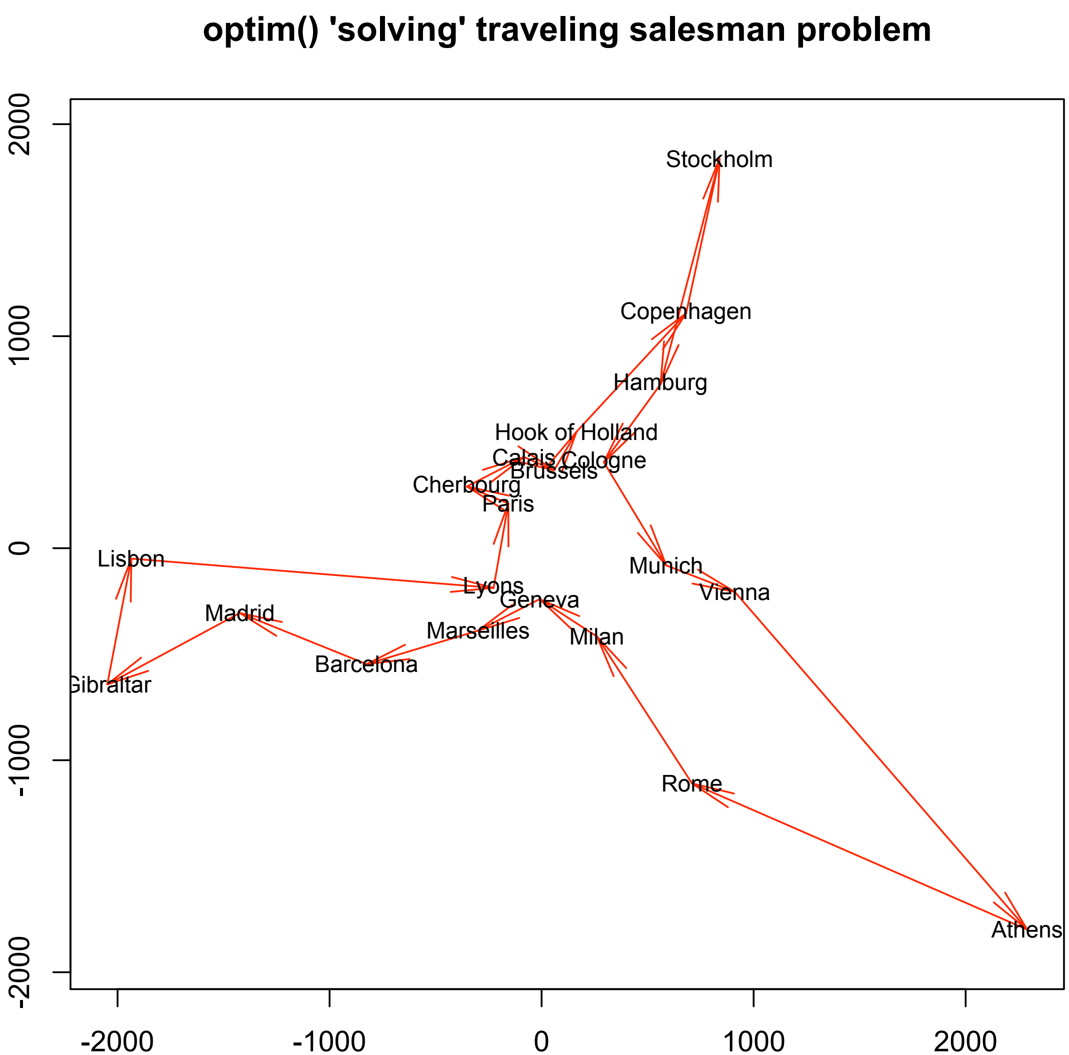
III. optim will do this for functions where you manipulate one vector (which can of course actually be a matrix)

# optim



optim() minimising 'wild function'
fw (x)
160
140
120
100
80
-40
-20
0
20
40
x


initial solution of traveling salesman problem
Stockholm
Copenhagen
Hamburg
Hook of Holland
Calais
Cologne
Brussels
Cherbourg
Paris
Lisbon
Munich
Lyons
Geneva
Vienna
Madrid
Marseilles
Milan
Barcelona
Gibraltar
Rome
Athens
2000
1000
0
-1000
-2000
-2000
-1000
0
1000
2000


optim() 'solving' traveling salesman problem
Stockholm
Copenhagen
Hamburg
Hook of Holland
Calais
Cologne
Brussels
Cherbourg
Paris
Lisbon
Lyons
Geneva
Munich
Vienna
Madrid
Marseilles
Milan
Barcelona
Gibraltar
Rome
Athens
2000
1000
0
-1000
-2000
-2000
-1000
0
1000
2000

# Trying to make a new function to do OLS FA

I. First, look at current ML FA function

   A. factanal

   B. It turns out that the critical optimization is done in factanal.fit.mle, but where is that?

II. getAnywhere(factanal.fit.mle)

   A. then look at the code

   B. scratch your head and try running it

# Sharing your code

I. Post the source code on your web site

II. develop a package which you keep on a “repository” on your web site

III. develop a package and upload it to CRAN

# Package development

I. a somewhat dated tutorial is found at http://personality-project.org/r/makingpackages.html

# Package development

I. package.skeleton(yourpackage) #creates directories and subdirectories for a package

   A. this includes a number of different subdirectories and files.

II. prompt(yourfunction) #creates a draft help file for your function

III. Document as you go

# in X11

I. R CMD check mypackage   #makes sure all the code is correct, checks for matches with documentation, runs all the example,

II. R CMD make mypackage #builds the package, but does not check except for working R

   A. this will convert the helpfiles from Rd files to html, LaTeX, and pdf

III. R CMD install mypackage will add any changes to your current package

# Using X11 to check packages

```
xterm
bash-3.2$ cd /Volumes/Test
bash-3.2$ R CMD check psych
* checking for working pdflatex ... OK
* using log directory '/Volumes/Test/psych.Rcheck'
* using R version 2.9.2 Patched (2009-09-24 r49893)
* using session charset: ASCII
* checking for file 'psych/DESCRIPTION' ... OK
* this is package 'psych' version '1.0-82'
* checking package name space information ... OK
* checking package dependencies ... OK
* checking if this is a source package ... OK
* checking for executable files ... OK
* checking whether package 'psych' can be installed ... OK
* checking package directory ... OK
* checking for portable file names ... OK
* checking for sufficient/correct file permissions ... OK
* checking DESCRIPTION meta-information ... OK
* checking top-level files ... OK
* checking index information ... OK
* checking package subdirectories ... OK
* checking R files for non-ASCII characters ... OK
* checking R files for syntax errors ... OK
* checking whether the package can be loaded ... OK
* checking whether the package can be loaded with stated dependencies ... OK
* checking whether the name space can be loaded with stated dependencies ... OK
* checking for unstated dependencies in R code ... OK
* checking S3 generic/method consistency ... OK
* checking replacement functions ... OK
* checking foreign function calls ... OK
* checking R code for possible problems ... OK
* checking Rd files ... OK
* checking Rd files against version 2 parser ... OK
* checking Rd cross-references ... OK
* checking for missing documentation entries ... OK
* checking for code/documentation mismatches ... OK
* checking Rd \usage sections ... OK
* checking data for non-ASCII characters ... OK
* checking examples ... OK
* checking package vignettes in 'inst/doc' ... OK
* checking PDF version of manual ... OK

bash-3.2$ 
```

# The structure of a package

| | | | |
|---|---|---|---|
| psych.Rcheck | Today, 3:16 PM | -- | Folder |
| psych-manual.pdf | Today, 3:15 PM | 1 MB | Adobe PDF document |
| psych-manual.log | Today, 3:15 PM | 70 KB | BBEdit text document |
| 00check.log | Today, 3:15 PM | 4 KB | BBEdit text document |
| psych-Ex.Rout | Today, 3:14 PM | 180 KB | Document |
| psych-Ex.ps | Today, 3:14 PM | 6.5 MB | PostScript |
| inst | Today, 3:14 PM | -- | Folder |
| doc | Today, 3:14 PM | -- | Folder |
| test.ICLUST.graph.dot | Today, 3:13 PM | 4 KB | Microsoft Word document |
| psych-Ex.R | Today, 3:12 PM | 70 KB | BBEdit text document |
| R.css | Today, 3:12 PM | 4 KB | CSS style sheet |
| 00install.out | Today, 3:12 PM | 12 KB | gmon.out |
| psych | Today, 3:12 PM | -- | Folder |
| Meta | Today, 3:12 PM | -- | Folder |
| INDEX | Today, 3:12 PM | 16 KB | Document |
| doc | Today, 3:12 PM | -- | Folder |
| R-ex | Today, 3:12 PM | -- | Folder |
| latex | Today, 3:12 PM | -- | Folder |
| html | Today, 3:12 PM | -- | Folder |
| help | Today, 3:12 PM | -- | Folder |
| CONTENTS | Today, 3:12 PM | 25 KB | Document |
| R | Today, 3:11 PM | -- | Folder |
| man | Today, 3:11 PM | -- | Folder |
| NEWS | Today, 3:11 PM | 12 KB | SimpleText Format |
| DESCRIPTION | Today, 3:11 PM | 4 KB | Document |
| data | Today, 3:11 PM | -- | Folder |
| NAMESPACE | Today, 3:11 PM | 4 KB | Document |

# The package

| | | | |
|---|---|---|---|
| psych | Aug 28, 2009 11:02 AM | -- | Folder |
| CHANGES | Oct 5, 2009 9:48 AM | 12 KB | SimpleText Format |
| DESCRIPTION | Oct 5, 2009 9:48 AM | 8 KB | BBEdit text document |
| man | Oct 4, 2009 7:39 PM | -- | Folder |
| NAMESPACE | Sep 29, 2009 10:04 PM | 4 KB | SimpleText Format |
| R | Sep 27, 2009 6:34 PM | -- | Folder |
| sort.rowcol.R | Aug 28, 2009 11:02 AM | 4 KB | BBEdit text document |
| data | Aug 23, 2009 12:31 PM | -- | Folder |
| inst | Jun 30, 2009 7:39 PM | -- | Folder |
| doc | Yesterday, 2:16 PM | -- | Folder |
| psych_for_sem.pdf | Yesterday, 2:16 PM | 414 KB | Adobe PDF document |
| overview.pdf | Yesterday, 2:15 PM | 2.3 MB | Adobe PDF document |
| psych_for_sem | Sep 30, 2009 9:25 AM | 37 KB | Rnw File |
| overview | Sep 30, 2009 9:00 AM | 66 KB | Rnw File |
| psych_manual.pdf | Jul 26, 2009 12:23 PM | 1 MB | Adobe PDF document |
| all | Jun 30, 2009 7:36 PM | 1.5 MB | BibTeX |
| apa.bst | Mar 21, 2007 4:33 PM | 25 KB | bst File |
| NEWS | Sep 30, 2009 7:45 PM | 12 KB | SimpleText Format |

# R CMD build psych

```
  mat.sort                         text    html    latex   example
  matrix.addition                  text    html    latex   example
  msq                              text    html    latex   example
  multi.hist                       text    html    latex   example
  neo                              text    html    latex   example
  omega                            text    html    latex   example
  omega.graph                      text    html    latex   example
  p.rep                            text    html    latex   example
  paired.r                         text    html    latex   example
  pairs.panels                     text    html    latex   example
  partial.r                        text    html    latex   example
  peas                             text    html    latex   example
  phi                              text    html    latex   example
  phi.demo                         text    html    latex   example
  phi2poly                         text    html    latex   example
  plot.psych                       text    html    latex   example
  polar                            text    html    latex   example
  poly.mat                         text    html    latex   example
  polychor.matrix                  text    html    latex   example
  principal                        text    html    latex   example
  print.psych                      text    html    latex   example
  r.test                           text    html    latex   example
  read.clipboard                   text    html    latex   example
  rescale                          text    html    latex   example
  sat.act                          text    html    latex   example
  scaling.fits                     text    html    latex
  schmid                           text    html    latex   example
  score.alpha                      text    html    latex   example
  score.items                      text    html    latex   example
  score.multiple.choice            text    html    latex   example
  sim                              text    html    latex   example
  sim.VSS                          text    html    latex   example
  sim.anova                        text    html    latex   example
  sim.congeneric                   text    html    latex   example
  sim.hierarchical                 text    html    latex   example
  sim.item                         text    html    latex   example
  sim.structural                   text    html    latex   example
  simulation.circ                  text    html    latex   example
  skew                             text    html    latex   example
  smc                              text    html    latex   example
  structure.graph                  text    html    latex   example
  structure.list                   text    html    latex   example
  super.matrix                     text    html    latex   example
  table2matrix                     text    html    latex   example
  test.psych                       text    html    latex   example
  thurstone                        text    html    latex   example
  tr                               text    html    latex   example
  vegetables                       text    html    latex   example
  winsor                           text    html    latex   example
** building package indices ...
* DONE (psych)
* creating vignettes ... OK
* removing junk files
* excluding invalid files from 'psych'
Subdirectory 'R' contains invalid file names:
  correlations neo pearson power reliability.data.sets
* checking for LF line-endings in source and make files
* checking for empty or unneeded directories
* building 'psych_1.0-82.tar.gz'
```

# R in the classroom

I. Undergraduate statistics and research methods

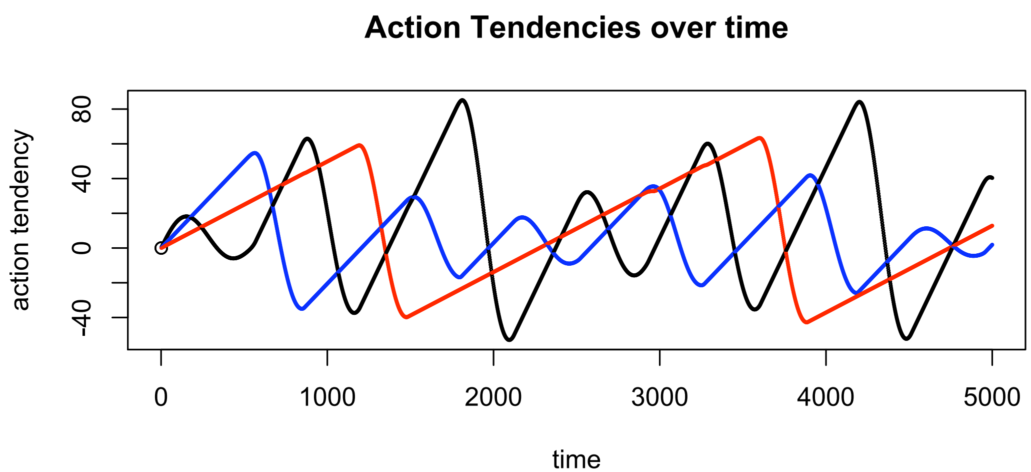
   A. describe, pairs.panels, anova, lm

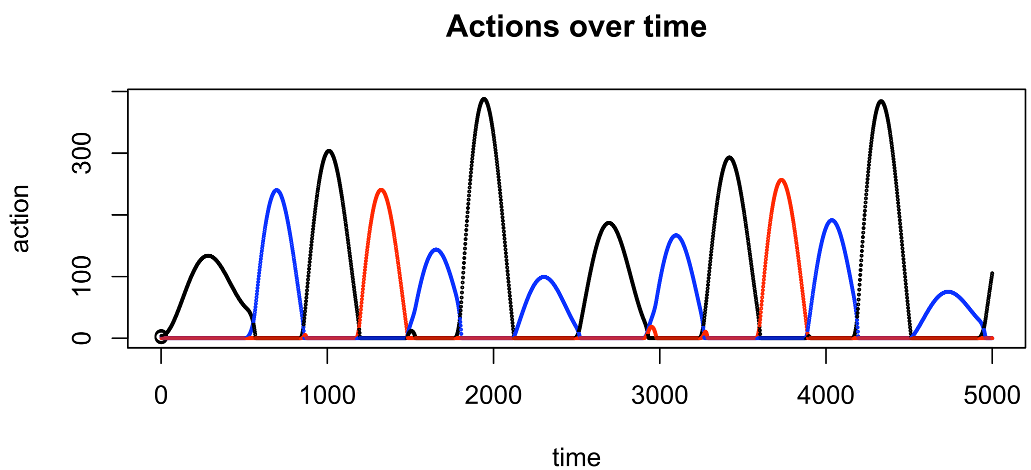
   B. plot, curve, etc.

   C. see tutorials for 205 and 371

   D. simulations of data for simulated studies

   E. Examples of complex models


Action Tendencies over time
action tendency
80
40
0
-40
0
1000
2000
3000
4000
5000
time


Actions over time
action
300
100
0
0
1000
2000
3000
4000
5000
time

# R in the classroom

I. Graduate

   A. data simulations

   B. data analysis

   C. longer tutorial