Psychology 350: An introduction to R for Psychological Research
Week 3b Iterative fitting

William Revelle
Northwestern University
Evanston, Illinois USA

https://personality-project.org/courses/350

Spring, 2024

Outline

Fitting functions as way of testing theory

1. Closed form versus open form.
2. Much of classical statistics is "closed form". That is, it is a matter of solving some equations using basic algebra.
   - Examples include the
   - t.test
   - F. test
   - basic linear regression
3. However, other functions are "open form" and have to be estimated using the process of iteration.
4. Although computers are useful for solving closed form equations, they are particularly useful for solving open forms.

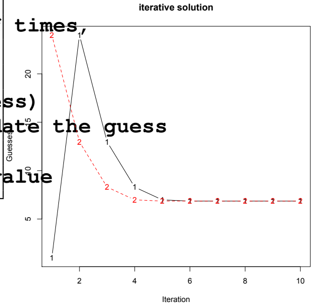
The basic concept of fitting

1. Given some fit statistic, try to find the optimal values for that fit
   - The Newton-Raphson method
     - Uses a linear approximation to the function (f) we are trying to solve
     - Get an initial estimate $x_0 = f(0)$
     - Find the first derivative at that point $f'(x_0)$
     - New estimate $= x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$
   - Consider the case of the square root of 47
     - Guess X
     - find result $= 47/Guess$
     - try a new $Guess = 47/(Guess + result)/2$
     - do it again
2. The next example is a baby function to do this
3. It is the concept, not the programming that is important

Iterative fitting to find a square root

R code

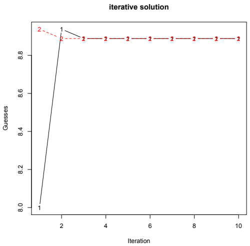
```
iterative.sqrt <- function(X,guess) {
if(missing(guess)) guess <- 1  #a dumb guess
iterate <- function(X,guess) { #another small function
    low <- guess
    high <- X/guess
    guess <-((high+low)/2)
    guess}
Iter <- matrix(NA,ncol=3,nrow=10)
for (i in 1:10) {  #do a for fixed number of times,
        or could set a stopping criterion
Iter[i,1] <- guess
Iter[i,2] <- error <-  guess - iterate(X,guess)
Iter[i,3] <- guess <- iterate(X, guess) #update the guess
}  #of if loop
Iter}   # end of function and  returns the value
```




```
> X <- 47
> iter <- iterative.sqrt(47)
> iter
 iterative.sqrt(47)
          [,1]          [,2]      [,3]
 [1,]  1.000000 -2.300000e+01 24.000000
 [2,] 24.000000  1.102083e+01 12.979167
 [3,] 12.979167  4.678989e+00  8.300177
 [4,]  8.300177  1.318824e+00  6.981353
```

Iterative fitting to find a square root – a better guess

```
> X <- 79
> iter <- iterative.sqrt(X,8)
> iter
           [,1]          [,2]     [,3]
 [1,] 8.000000 -9.375000e-01 8.937500
 [2,] 8.937500  4.916958e-02 8.888330
 [3,] 8.888330  1.360012e-04 8.888194
 [4,] 8.888194  1.040501e-09 8.888194
 [5,] 8.888194  0.000000e+00 8.888194
 [6,] 8.888194  0.000000e+00 8.888194
 [7,] 8.888194  0.000000e+00 8.888194
 [8,] 8.888194  0.000000e+00 8.888194
 [9,] 8.888194  0.000000e+00 8.888194
[10,] 8.888194  0.000000e+00 8.888194
```




```
> matplot(iter,typ="b",main="iterative solution",xlab="Iteration",
          ylab="Guesses")
```

Control the number of iterations

R code

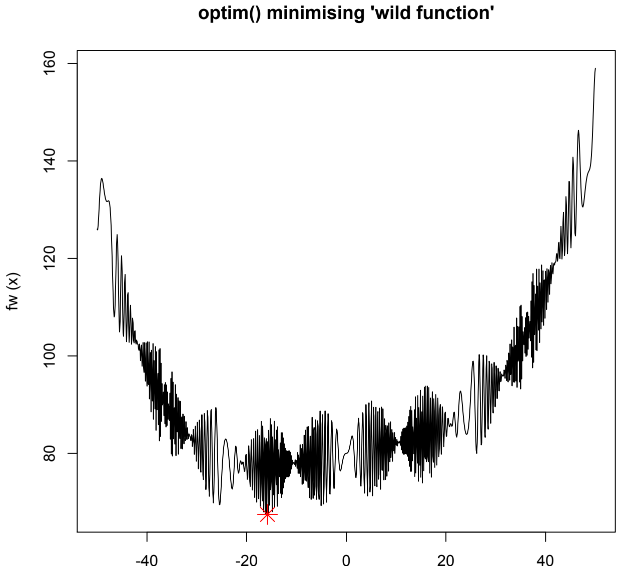
```
iterative.sqrt <- function(X,guess,n.iter=10) {
if(missing(guess)) guess <- 1  #a dumb guess

iterate <- function(X,guess) {   #  another small function
    low <- guess
    high <- X/guess
    guess <-((high+low)/2)
    guess}
Iter <- matrix(NA,ncol=3,nrow=n.iter)
for (i in 1:n.iter) {  #do a for fixed number of times,
               or could set a stopping criterion

Iter[i,1] <- guess
Iter[i,2] <- error <-  guess - iterate(X,guess)  #
Iter[i,3] <- guess <- iterate(X, guess) #update the guess

}
Iter}   #this returns the value
```

The optim function is one way to fit complex models



Iterative fitting has several steps

1. Specifying a model and loss function
   - Ideally supplying a first derivative to the loss function
   - Can be done empirically
2. Some way to evaluate the quality of the fit
3. Minimization of function, but then how good is this minimum?
4. Goodness of fit statistics are reported for the optimal value.
5. Examples of using the `optimin` function include `tetrachoric`, `polychoric` and `fa`

Multivariate analysis

- Many procedures use this concept of iterative fitting
- Some, such as principal components are "closed form" and can just be solved directly
- Others, such as "factor analysis" need to find solutions by successive approximations
- The issue of factor or component rotation is an iterative procedure.
- Principal Components and Factor analysis are all the result of some basic matrix equations to approximate a matrix
- Conceptually, we are just taking the square root of a correlation matrix:
- $R \approx CC'$ or $R \approx FF' + U^2$ (u^2 is a diagonal matrix)
- For any given correlation matrix, R, can we find a C or an F matrix which approximates it?

Consider the following matrix

What would be its “square root”? That is to say, what simpler matrix, times itself is equal to R?

```
R
     V1   V2   V3   V4
V1 1.00 0.56 0.48 0.40
V2 0.56 1.00 0.42 0.35
V3 0.48 0.42 1.00 0.30
V4 0.40 0.35 0.30 1.00
```

Find $R \approx FF' + U^2$

1. Find an eigen value decomposition of R: $\mathbf{R} = \mathbf{X}\lambda\mathbf{X'}$
2. Principal Components are just another way of expressing the eigen value decomposition
3. $C = \mathbf{X}\sqrt{\lambda}$ and $\mathbf{R} = \mathbf{CC}'$
4. Take just the first n columns of $\mathbf{X}$ and λ
5. $\mathbf{C}_n = \mathbf{X_{1:n}}\sqrt{\lambda_{1:n}}$ and $\mathbf{R} \approx \mathbf{C_n}\ \mathbf{_nC}$

Principal Components

```
R                                               The original correlation matrix
     V1   V2   V3   V4
V1 1.00 0.56 0.48 0.40
V2 0.56 1.00 0.42 0.35
V3 0.48 0.42 1.00 0.30
V4 0.40 0.35 0.30 1.00
```

R code

```
P1 <- pca(R)
loadings <- P1$loadings
round(loadings,2)
model<- loadings %*% t(loadings)
round(model,2)
residual(P1)
```

```
round(model,2)                                   This is the predicted R given the loadings
     V1   V2   V3   V4
V1 0.69 0.65 0.61 0.54
V2 0.65 0.62 0.58 0.51
V3 0.61 0.58 0.53 0.48
V4 0.54 0.51 0.48 0.43

resid(P1)                                           These are the residuals
   V1    V2    V3    V4
V1  0.31
V2 -0.09  0.38
V3 -0.13 -0.16  0.47
V4 -0.14 -0.16 -0.18  0.57
```

We create this matrix using the `sim.congeneric` function

First, create an example matrix

R code

```
"sim.congeneric" <-  function{
 (loads = c(0.8, 0.7, 0.6, 0.5),...)  {
..
n <- length(loads)
 loading <- matrix(loads, nrow = n)
error <- diag(1, nrow = n)
diag(error) <- sqrt(1 - loading^2)

 model <- pattern %*% t(pattern)

...
 result <- model
 return(result)
 }
```

1. Give some default values
2. Create a correlation matrix using matrix algebra
   - $model = \mathbf{FF'}$
   - $\text{diag(model)} = 1$
3. Return the value

A congeneric matrix is one with just one factor

R code

```
R <- sim.congeneric()
R
f1 <- fa(R)
```

```
> R
     V1   V2   V3   V4
V1 1.00 0.56 0.48 0.40
V2 0.56 1.00 0.42 0.35
V3 0.48 0.42 1.00 0.30
V4 0.40 0.35 0.30 1.00
> f1 <- fa(R)
> f1
Factor Analysis using method =  minres
Call: fa(r = R)
Standardized loadings (pattern matrix) based upon correlation matrix
   MR1   h2   u2 com
V1 0.8 0.64 0.36   1
V2 0.7 0.49 0.51   1
V3 0.6 0.36 0.64   1
V4 0.5 0.25 0.75   1

                MR1
SS loadings    1.74
Proportion Var 0.43
```

But does this really fit?

Think about the residuals (Data - model)

R code

```
R
round(f1$model,2)
residuals(f1)
```

```
> R
     V1   V2   V3   V4
V1 1.00 0.56 0.48 0.40
V2 0.56 1.00 0.42 0.35
V3 0.48 0.42 1.00 0.30
V4 0.40 0.35 0.30 1.00
> round(f1$model,2)
     V1   V2   V3   V4
V1 0.64 0.56 0.48 0.40
V2 0.56 0.49 0.42 0.35
V3 0.48 0.42 0.36 0.30
V4 0.40 0.35 0.30 0.25
> residuals(f1)
   V1   V2   V3   V4
V1 0.36
V2 0.00 0.51
V3 0.00 0.00 0.64
V4 0.00 0.00 0.00 0.75
```

Hence we add the fudge factor U2 = diagonal of residuals and say
$R = FF' + U2$

How did we find F: The OLS function

Function hidden inside of the fa function

R code

```
FA.OLS <- function(Psi,S,nf) {
     E <- eigen(S-diag(Psi),symmetric=T)
     U <-E$vectors[,1:nf,drop=FALSE]
     D <- E$values[1:nf,drop=FALSE]
     D [D < 0] <- 0
        if(length(D) < 2) {L <- U * sqrt(D)} else {
              L <- U %*% diag(sqrt(D))} #gets around a weird problem :
             model <- L %*% t(L)
     diag(model) <- diag(S)
     return(sum((S-model)^2)/2)
 }
```

```
#try different values of psi to see which fits the best
psi <- c(1,1,1,1)
>  FA.OLS(psi,R,1)
[1] 0.08195686
> psi <- smc(R)
> FA.OLS(psi,R,1)
[1] 0.05153827
> psi <- c(.64,.49,.36,.25)
> FA.OLS(psi,R,1)
[1] 0.04983562
> psi <- c( 0.36 ,0.51, 0.64, 0.75)   #use these values to find out what the loadings are
> FA.OLS(psi,R,1)
[1] 1.787263e-31
```

Modeling the factors: The PFout function

R code

```
  FAout <- function(Psi, S, q) {
        sc <- diag(1/sqrt(Psi))
        Sstar <- sc %*% S %*% sc
        E <- eigen(Sstar, symmetric = TRUE)
        L <- E$vectors[, 1L:q, drop = FALSE]
        load <- L %*% diag(sqrt(pmax(E$values[1L:q] - 1, 0)),
            q)
        diag(sqrt(Psi)) %*% load
   }

loadings <- FAout(psi,R,1) #psi and R from previous solution
t(loadings)#  show them
model <- loadings %*% t(loadings)
```

```
 t(loadings)
     [,1] [,2] [,3] [,4]
[1,]  0.8  0.7  0.6  0.5

model
     [,1] [,2] [,3] [,4]
[1,] 0.64 0.56 0.48 0.40
[2,] 0.56 0.49 0.42 0.35
[3,] 0.48 0.42 0.36 0.30
[4,] 0.40 0.35 0.30 0.25
```

If X = a data matrix of n variables for N subjects

1. Do we really want n separate scores?
2. Can we find (far) fewer variables that account for the scores?
3. "Accounting" for the scores says can we model the scores (or at least their correlations)?
4. Components account for the most variance in the scores.
5. Factors are models of the correlations/covariances of the scores.
6. When we find "scale score" by summing or averaging the items we are saying that the items all measure one thing, but do not correlate perfectly because of random error.
7. This is the model of reliability theory.

Several example data sets in psychTools

- `ability` 16 items measuring cognitive ability from (Condon & Revelle, 2014) and described more fully in (Revelle, Dworak & Condon, 2020) and (Dworak, Revelle, Doebler & Condon, 2020).
- `bfi` 25 personality items and 3 demographic variables from the SAPA project. The 25 items are thought to represent the "Big 5" CANOE dimensions.
- `sai` 20 State Anxiety Items and 3 demographics from the PMC lab. Discussed in greater detail in (Revelle & Condon, 2019).
- `msqR` 75 mood and 13 condition variables from the Motivational State Questionnaire collected in the PMC lab.
- `spi` 135 personality items and 10 demographic variables from the SAPA Personality Questionnaire (Condon, 2018) collected using the SAPA-project.org web site.

The `ability` data set

1. 16 multiple choice ability items 1525 subjects taken from the Synthetic Aperture Personality Assessment (SAPA) web based personality assessment project are saved as iqitems. Those data are shown as examples of how to score multiple choice tests and analyses of response alternatives. When scored correct or incorrect, the data are useful for demonstrations of tetrachoric based factor analysis irt.fa and finding tetrachoric correlations.
2. 16 items were sampled from 80 items given as part of the SAPA project (Revelle, Wilt & Rosenthal, 2010) to develop online measures of ability. These 16 items reflect four lower order factors (verbal reasoning, letter series, matrix reasoning, and spatial rotations. These lower level factors all share a higher level factor ('g').

bfi

1. 25 personality self report items taken from the International Personality Item Pool (ipip.ori.org) were included as part of the Synthetic Aperture Personality Assessment SAPA web based personality assessment project. The data from 2800 subjects are included here as a demonstration set for scale construction, factor analysis, and Item Response Theory analysis. Three additional demographic variables (sex, education, and age) are also included.
2. The first 25 items are organized by five putative factors: Agreeableness, Conscientiousness, Extraversion, Neuroticism, and Opennness. The scoring key is created using make.keys, the scores are found using score.items.
3. These five factors are a useful example of using irt.fa to do Item Response Theory based latent factor analysis of the polychoric correlation matrix. The endorsement plots for each item, as well as the item information functions reveal that the items differ in their quality.
4. The item data were collected using a 6 point response scale: 1 Very Inaccurate 2 Moderately Inaccurate 3 Slightly Inaccurate 4 Slightly Accurate 5 Moderately Accurate 6 Very Accurate.

sai

1. State Anxiety was measured two-three times in 11 studies at the Personality-Motivation-Cognition laboratory. Here are item responses for 11 studies (9 repeated twice, 2 repeated three times). In all studies, the first occasion was before a manipulation. In some studies, caffeine, or movies or incentives were then given to some of the participants before the second and third STAI was given. In addition, Trait measures are available and included in the tai data set (3032 subjects).
2. The standard experimental study at the Personality, Motivation and Cognition (PMC) laboratory (Revelle & Anderson, 1998) was to administer a number of personality trait and state measures (e.g. the epi, msq, msqR and sai) to participants before some experimental manipulation of arousal/effort/anxiety. Following the manipulation (with a 30 minute delay if giving caffeine/placebo), some performance task was given, followed once again by measures of state arousal/effort/anxiety.
3. Here are the item level data on the sai (state anxiety) and the tai (trait anxiety). Scores on these scales may be found using the scoring keys. The affect data set includes pre and post scores for two studies (flat and maps) which manipulated state by using four types of movies.

msqR

1. Emotions may be described either as discrete emotions or in dimensional terms. The Motivational State Questionnaire (MSQ) was developed to study emotions in laboratory and field settings. The data can be well described in terms of a two dimensional solution of energy vs tiredness and tension versus calmness. Alternatively, this space can be organized by the two dimensions of Positive Affect and Negative Affect. Additional items include what time of day the data were collected and a few personality questionnaire scores. 3032 unique participants took the MSQ at least once, 2753 at least twice, 446 three times, and 181 four times. The 3032 participants also took the sai state anxiety inventory at the same time. Some studies manipulated arousal by caffeine, others manipulations included affect inducing movies.
2. The Motivational States Questionnaire (MSQ) is composed of 75 items, which represent the full affective space (Revelle & Anderson, 1998). The MSQ consists of 20 items taken from the Activation-Deactivation Adjective Check List (Thayer, 1989), 18 from the Positive and Negative Affect Schedule (PANAS,0(Watson, Clark & Tellegen, 1988) along with the affective circumplex items used by Larsen & Diener (1992) The response format was a four-point scale that corresponds to Russell & Carroll (1999) "ambiguous–likely-unipolar format" and that asks the respondents to indicate their current standing ("at this moment") with the following rating scale:
   0————————-1————————-2————————-3
   Not at all A little Moderately Very much

spi

1. The SPI (SAPA Personality Inventory) is a set of 135 items primarily selected from the International Personality Item Pool (ipip.ori.org). This is an example data set collected using SAPA procedures at the sapa-project.org web site. This data set includes 10 demographic variables as well. The data set with 4000 observations on 145 variables may be used for examples in scale construction and validation, as well as empirical scale construction to predict multiple criteria.
2. Using the data contributed by about 125,000 visitors to the https://SAPA-project.org website, David Condon has developed a hierarchical framework for assessing personality at two levels. The higher level has the familiar five factors that have been studied extensively in personality research since the 1980s – Conscientiousness, Agreeableness, Neuroticism, Openness, and Extraversion. The lower level has 27 factors that are considerably more narrow. These were derived based on administrations of about 700 public-domain IPIP items to 3 large samples. Condon describes these scales as being "empirically-derived" because relatively little theory was used to select the number of factors in the hierarchy and the items in the scale for each factor (to be clear, he means relatively little personality theory though he relied on quite a lot of sampling and statistical theory). You can read all about the procedures used to develop this framework in his book/manual. If you would like to reproduce these analyses, you can download the data files from Dataverse (links are also provided in the manual) and compile this script in R (he used knitR). Instructions are provided in the Preface to the manual.
3. This small subset of the data is provided for demonstration purposes.

The ability data set

R code

```
dim(ability)
describe(ability)
lowerCor(ability)
```

```
dim(ability)
[1] 1525   16
> dim(ability)
[1] 1525   16
> describe(ability)
           vars    n mean   sd median trimmed mad min max range  skew kurtosis   se
reason.4      1 1442 0.68 0.47      1    0.72   0   0   1     1 -0.75    -1.44 0.01
reason.16     2 1463 0.73 0.45      1    0.78   0   0   1     1 -1.02    -0.96 0.01
reason.17     3 1440 0.74 0.44      1    0.80   0   0   1     1 -1.08    -0.84 0.01
reason.19     4 1456 0.64 0.48      1    0.68   0   0   1     1 -0.60    -1.64 0.01
letter.7      5 1441 0.63 0.48      1    0.67   0   0   1     1 -0.56    -1.69 0.01
letter.33     6 1438 0.61 0.49      1    0.63   0   0   1     1 -0.43    -1.82 0.01
letter.34     7 1455 0.64 0.48      1    0.68   0   0   1     1 -0.59    -1.65 0.01
letter.58     8 1438 0.47 0.50      0    0.46   0   0   1     1  0.12    -1.99 0.01
matrix.45     9 1458 0.55 0.50      1    0.56   0   0   1     1 -0.20    -1.96 0.01
matrix.46    10 1470 0.57 0.50      1    0.59   0   0   1     1 -0.28    -1.92 0.01
matrix.47    11 1465 0.64 0.48      1    0.67   0   0   1     1 -0.57    -1.67 0.01
matrix.55    12 1459 0.39 0.49      0    0.36   0   0   1     1  0.45    -1.80 0.01
rotate.3     13 1456 0.20 0.40      0    0.13   0   0   1     1  1.48     0.19 0.01
rotate.4     14 1460 0.22 0.42      0    0.15   0   0   1     1  1.34    -0.21 0.01
rotate.6     15 1456 0.31 0.46      0    0.27   0   0   1     1  0.80    -1.35 0.01
rotate.8     16 1460 0.19 0.39      0    0.12   0   0   1     1  1.55     0.41 0.01
```

ability: descriptives, continued

```
 lowerCor(ability)
           rsn.4 rs.16 rs.17 rs.19 ltt.7 lt.33 lt.34 lt.58 mt.45 mt.46 mt.47 mt.55 rtt.3 rtt.4
reason.4   1.00
reason.16  0.28  1.00
reason.17  0.40  0.32  1.00
reason.19  0.30  0.25  0.34  1.00
letter.7   0.28  0.27  0.29  0.25  1.00
letter.33  0.23  0.20  0.26  0.25  0.34  1.00
letter.34  0.29  0.26  0.29  0.27  0.40  0.37  1.00
letter.58  0.29  0.21  0.29  0.25  0.33  0.28  0.32  1.00
matrix.45  0.25  0.18  0.20  0.22  0.20  0.20  0.21  0.19  1.00
matrix.46  0.25  0.18  0.24  0.18  0.24  0.23  0.27  0.21  0.33  1.00
matrix.47  0.24  0.24  0.27  0.23  0.27  0.23  0.30  0.23  0.24  0.23  1.00
matrix.55  0.16  0.15  0.16  0.15  0.14  0.17  0.14  0.23  0.21  0.14  0.21  1.00
rotate.3   0.23  0.16  0.17  0.18  0.18  0.17  0.19  0.24  0.16  0.15  0.20  0.18  1.00
rotate.4   0.25  0.20  0.20  0.21  0.23  0.21  0.21  0.27  0.17  0.17  0.20  0.18  0.53  1.00
rotate.6   0.25  0.20  0.27  0.19  0.20  0.21  0.19  0.26  0.15  0.20  0.18  0.17  0.43  0.45
rotate.8   0.21  0.16  0.18  0.16  0.13  0.14  0.15  0.22  0.16  0.15  0.17  0.19  0.43  0.44
>
```

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

One way to test techniques is simulation

1. With real data, we do not know 'truth'
2. With simulated data, we can know the right answer
3. Lets examine the sim item function
   - defaults to certain parameter values
   - processes the data (make it up)
   - returns values
4. Simulations can default to certain values but allow you to specify other values

sim.item Simulates a 2 dimensional item structure

```
function (nvar = 72, nsub = 500, circum = FALSE, xloading = 0.6,
    yloading = 0.6, gloading = 0, xbias = 0, ybias = 0, categorical = FALSE,
    low = -3, high = 3, truncate = FALSE, threshold = NULL)      {
    avloading <- (xloading + yloading)/2
    errorweight <- sqrt(1 - (avloading^2 + gloading^2))
    g <- rnorm(nsub)
    truex <- rnorm(nsub) * xloading + xbias
    truey <- rnorm(nsub) * yloading + ybias
    if (!is.null(threshold)) {
        if (length(threshold) < nvar)
            threshold <- sample(threshold, nvar, replace = TRUE)      }
    if (circum) {
        radia <- seq(0, 2 * pi, len = nvar + 1)
        rad <- radia[which(radia < 2 * pi)]
            }  else {rad <- c(rep(0, nvar/4), rep(pi/2, nvar/4), rep(pi,
        nvar/4), rep(3 * pi/2, nvar/4))}
    error <- matrix(rnorm(nsub * (nvar)), nsub)
    trueitem <- outer(truex, cos(rad)) + outer(truey, sin(rad))
    item <- gloading * g + trueitem + errorweight * error
    if (categorical) {
        if (is.null(threshold)) {
            item = round(item)
            item[(item <= low)] <- low
            item[(item > high)] <- high
        }   else {
            i <- 1:nvar
            item <- t(t(item[, i]) > threshold[i]) + 0
        }}
    colnames(item) <- paste("V", 1:nvar, sep = "")
    return(item)
}
```

Simulate 2 factor data for 12 variables

Using the sim.item function

```
 my.data <- sim.item(12, categorical =TRUE)  #make them look like items
my.cor <- cor(my.data)
round(my.cor,2)
       V1    V2    V3    V4    V5    V6    V7    V8    V9   V10   V11   V12
V1   1.00  0.32  0.35 -0.02  0.04  0.05 -0.33 -0.38 -0.37  0.02  0.05  0.00
V2   0.32  1.00  0.33 -0.06 -0.01  0.03 -0.33 -0.31 -0.33  0.08  0.02  0.01
V3   0.35  0.33  1.00 -0.03  0.00 -0.01 -0.38 -0.33 -0.27  0.01  0.04 -0.11
V4  -0.02 -0.06 -0.03  1.00  0.29  0.36 -0.08  0.01  0.00 -0.31 -0.33 -0.28
V5   0.04 -0.01  0.00  0.29  1.00  0.33  0.00  0.07  0.01 -0.29 -0.30 -0.28
V6   0.05  0.03 -0.01  0.36  0.33  1.00 -0.06  0.10 -0.03 -0.37 -0.33 -0.28
V7  -0.33 -0.33 -0.38 -0.08  0.00 -0.06  1.00  0.30  0.30 -0.01  0.02  0.10
V8  -0.38 -0.31 -0.33  0.01  0.07  0.10  0.30  1.00  0.36 -0.10 -0.11  0.02
V9  -0.37 -0.33 -0.27  0.00  0.01 -0.03  0.30  0.36  1.00 -0.05 -0.02  0.00
V10  0.02  0.08  0.01 -0.31 -0.29 -0.37 -0.01 -0.10 -0.05  1.00  0.40  0.33
V11  0.05  0.02  0.04 -0.33 -0.30 -0.33  0.02 -0.11 -0.02  0.40  1.00  0.35
V12  0.00  0.01 -0.11 -0.28 -0.28 -0.28  0.10  0.02  0.00  0.33  0.35  1.00
```

Remember to always describe your data

R code

```
describe(my.data)
```

```
describe(my.data)
    vars   n  mean   sd median trimmed  mad min max range  skew kurtosis   se
V1     1 500  0.00 1.05      0    0.00 1.48  -3   3     6  0.06     0.03 0.05
V2     2 500  0.00 1.03      0    0.00 1.48  -3   3     6  0.05    -0.08 0.05
V3     3 500 -0.04 1.07      0   -0.06 1.48  -3   3     6  0.15    -0.23 0.05
V4     4 500 -0.06 1.02      0   -0.04 1.48  -3   3     6 -0.12     0.16 0.05
V5     5 500 -0.04 0.98      0   -0.07 1.48  -3   3     6  0.13    -0.19 0.04
V6     6 500 -0.05 1.06      0   -0.05 1.48  -3   3     6  0.07     0.01 0.05
V7     7 500  0.01 1.01      0    0.02 1.48  -3   3     6 -0.16     0.13 0.05
V8     8 500 -0.01 1.08      0   -0.01 1.48  -3   3     6 -0.04    -0.03 0.05
V9     9 500  0.03 1.06      0    0.02 1.48  -3   3     6  0.03    -0.24 0.05
V10   10 500  0.04 1.02      0    0.07 1.48  -3   2     5 -0.20    -0.16 0.05
V11   11 500  0.03 1.03      0    0.01 1.48  -3   3     6 -0.01    -0.06 0.05
V12   12 500 -0.03 0.98      0   -0.01 1.48  -3   3     6 -0.11     0.11 0.04
>
```

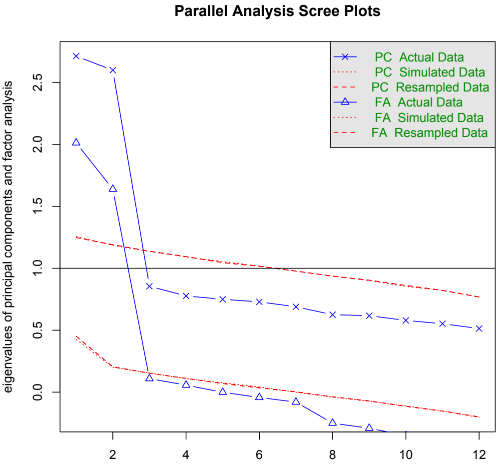
Multiple ways to determine how many factors are in the data

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 & 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.data)
fa.parallel(my.data)
Parallel analysis suggests that the number of factors =  2  and the number of components =  2
```



Take out 2 factors from my.cor

R code

```
f2 <- fa(my.cor,nfactors =2)
```

```
Factor Analysis using method =  minres
Call: fa(r = my.cor, nfactors = 2)
Standardized loadings (pattern matrix) based upon correlation matrix
      MR1   MR2   h2   u2 com
V1    0.64 -0.02 0.41 0.59 1.0
V2    0.59  0.02 0.35 0.65 1.0
V3    0.61 -0.04 0.37 0.63 1.0
V4    0.03 -0.58 0.34 0.66 1.0
V5    0.01 -0.55 0.30 0.70 1.0
V6    0.03 -0.60 0.36 0.64 1.0
V7   -0.58  0.08 0.34 0.66 1.0
V8   -0.62 -0.10 0.39 0.61 1.1
V9   -0.59  0.00 0.35 0.65 1.0
V10   0.07  0.61 0.38 0.62 1.0
V11   0.03  0.63 0.39 0.61 1.0
V12  -0.06  0.57 0.33 0.67 1.0

                       MR1  MR2
SS loadings           2.21 2.12
Proportion Var        0.18 0.18
Cumulative Var        0.18 0.36
Proportion Explained  0.51 0.49
Cumulative Proportion 0.51 1.00

 With factor correlations of
     MR1  MR2
MR1 1.00 0.04
MR2 0.04 1.00
```

2 factors of my.cor, continued

```
With factor correlations of
     MR1  MR2
MR1 1.00 0.04
MR2 0.04 1.00

Mean item complexity =  1
Test of the hypothesis that 2 factors are sufficient.

The degrees of freedom for the null model are  66  and the objective function was  2.52
The degrees of freedom for the model are 43  and the objective function was  0.11

The root mean square of the residuals (RMSR) is  0.03
The df corrected root mean square of the residuals is  0.03

Fit based upon off diagonal values = 0.99
Measures of factor score adequacy
                                                 MR1  MR2
Correlation of (regression) scores with factors   0.88 0.88
Multiple R square of scores with factors          0.78 0.77
Minimum correlation of possible factor scores     0.56 0.53
```

1 factor of ability, how well does it fit?

R code

```
f1 <- fa(ability)
```

```
> f1
Factor Analysis using method =  minres
Call: fa(r = ability)
Standardized loadings (pattern matrix) based upon correlation matrix
           MR1   h2   u2 com
reason.4  0.55 0.30 0.70   1
reason.16 0.45 0.20 0.80   1
reason.17 0.54 0.29 0.71   1
reason.19 0.47 0.22 0.78   1
letter.7  0.52 0.27 0.73   1
letter.33 0.48 0.23 0.77   1
letter.34 0.54 0.29 0.71   1
letter.58 0.53 0.28 0.72   1
matrix.45 0.41 0.17 0.83   1
matrix.46 0.43 0.18 0.82   1
matrix.47 0.47 0.22 0.78   1
matrix.55 0.35 0.12 0.88   1
rotate.3  0.50 0.25 0.75   1
rotate.4  0.55 0.30 0.70   1
rotate.6  0.53 0.28 0.72   1
rotate.8  0.46 0.21 0.79   1

                MR1
SS loadings    3.81
Proportion Var 0.24

Mean item complexity =  1
```

1 factor (continued)

```
The degrees of freedom for the null model are  120  and the objective function was
              3.28 with Chi Square of  4973.83
The degrees of freedom for the model are 104  and the objective function was  0.7

The root mean square of the residuals (RMSR) is  0.07
The df corrected root mean square of the residuals is  0.07

The harmonic number of observations is  1426 with the empirical chi square  1476.62
                with prob <  3e-241
The total number of observations was  1525  with Likelihood Chi Square =  1063.25
           with prob <  9.5e-159

Tucker Lewis Index of factoring reliability =  0.772
RMSEA index =  0.078  and the 90 % confidence intervals are  0.074 0.082
BIC =  300.96
Fit based upon off diagonal values = 0.93
Measures of factor score adequacy
                                                  MR1
Correlation of (regression) scores with factors   0.91
Multiple R square of scores with factors          0.84
Minimum correlation of possible factor scores     0.67
```

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.

Condon, D. M. & Revelle, W. (2014). The International Cognitive Ability Resource: Development and initial validation of a public-domain measure. *Intelligence*, *43*, 52–64.

Dworak, E. M., Revelle, W., Doebler, P., & Condon, D. M. (2020). Using the International Cognitive Ability Resource as an open source tool to explore individual differences in cognitive ability. *Personality and Individual Differences*.

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

Larsen, R. J. & Diener, E. (1992). Promises and problems with the circumplex model of emotion. In M. S. Clark (Ed.), *Emotion* (pp. 25–59). Thousand Oaks, CA: Sage Publications, Inc.

Revelle, W. & Anderson, K. J. (1998). Personality, motivation and cognitive performance: Final report to the army research institute on contract MDA 903-93-K-0008. Technical report, Northwestern University, Evanston, Illinois, USA.

Revelle, W. & Condon, D. M. (2019). Reliability: from alpha to omega. *Psychological Assessment*, *31*(12), 1395–1411.

Revelle, W., Dworak, E. M., & Condon, D. M. (2020). Cognitive ability in everyday life: the utility of open source measures. *Current Directions in Psychological Science*, *29*(4), 358–363.

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

Revelle, W., Wilt, J., & Rosenthal, A. (2010). Individual differences in cognition: New methods for examining the personality-cognition link. In A. Gruszka, G. Matthews, &

B. Szymura (Eds.), *Handbook of Individual Differences in Cognition: Attention, Memory and Executive Control* chapter 2, (pp. 27–49). New York, N.Y.: Springer.

Russell, J. A. & Carroll, J. M. (1999). On the bipolarity of positive and negative affect. *Psychological Bulletin*, *125*(1), 3–30.

Thayer, R. E. (1989). *The biopsychology of mood and arousal*. The biopsychology of mood and arousal. xi, 234 pp. New York, NY: Oxford University Press.

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

Watson, D., Clark, L. A., & Tellegen, A. (1988). Development and validation of brief measures of positive and negative affect: The PANAS scales. *Journal of Personality and Social Psychology*, *54*(6), 1063–1070.