Using R and the *psych* package to find ω

William Revelle
Department of Psychology
Northwestern University

December 6, 2021

Contents

1 Overview of this and related documents

To do basic and advanced personality and psychological research using R is not as complicated as some think. This is one of a set of "How To" to do various things using R (R Core Team, 2020), particularly using the *psych* (Revelle, 2020) package.

The current list of How To's includes:

1. An introduction (vignette) of the *psych* package

2. An overview (vignette) of the *psych* package

3. Installing R and some useful packages

4. Using R and the *psych* package to find *omega$_h$* and ω_t (this document)..

5. Using R and the *psych* for factor analysis and principal components analysis.

6. Using the `scoreItems` function to find scale scores and scale statistics.

7. Using `mediate` and `setCor` to do mediation, moderation and regression analysis

1.1 *omega$_h$* as an estimate of the general factor saturation of a test

Cronbach's coefficient *alpha* (Cronbach, 1951) is pehaps the most used (and most misused) estimate of the internal consistency of a test. α may be found in the *psych* package using the `alpha` function. However, two alternative estimates of reliability that take into account the hierarchical structure of the inventory are McDonald's ω_h and ω_t (McDonald, 1999). These may be found in R in one step using one of two functions in the *psych* package: the `omega` function for an exploratory analysis (See Figure 1) or `omegaSem` for a confirmatory analysis using the *sem* package solution based upon the exploratory solution from `omega`. This guide explains how to do it for the non or novice R user. These set of instructions are adapted from three different sets of notes that the interested reader might find helpful: A set of slides developed for a two hour short course in R given for several years to the Association of Psychological Science as well as a short guide to R for psychologists and the vignette for the *psych* package.

McDonald has proposed coefficient omega (hierarchical) (ω_h) as an estimate of the general factor saturation of a test. Zinbarg et al. (2005) and Revelle and Zinbarg (2009) compare compare McDonald's ω_h to Cronbach's α and Revelle's β. They conclude that ω_h is the best estimate. (See also Zinbarg et al. (2006) and Revelle and Zinbarg (2009) as well as Revelle and Condon (2018, 2019).

By following these simple guides, you soon will be able to do such things as find ω_h by issuing just three lines of code:

```
                          ┌─ R code ─┐
library(psych)
my.data <- read.file()
omega(my.data)
```

The resulting output will be both graphical and textual.

This guide helps the naive R user to issue those three lines. Be careful, for once you start using R, you will want to do more.

One way to find ω_h is to do a factor analysis of the original data set, rotate the factors obliquely, factor that correlation matrix, do a Schmid-Leiman (`schmid`) transformation to find general factor loadings, and then find ω_h. This is done using the `omega` function in the *psych* package in R. This requires installing and using both R as well as the *psych* package (Revelle, 2020).

1.1.1 But what about α?

Several statistics were developed in the 1930s-1950s as short cut estimates of reliability (Revelle and Condon, 2019). The approaches that consider just one test are collectively known as internal consistency procedures but also borrow from the concepts of domain sampling. Some of these techniques, e.g., Cronbach (1951); Guttman (1945); Kuder and Richardson (1937) were developed before advances in computational speed made it trivial to find the factor structure of tests, and were based upon test and item variances. These procedures (α, λ_3, KR20) were essentially short cuts for estimating reliability. To just find Guttman's λ_3 (Guttman, 1945) which is also known as *coefficient* α (Cronbach, 1951), you can use the `alpha` function or the `scoreItems` function. See the tutorial on how to use the `scoreItems` function to find scale scores and scale statistics. But, with modern computers, we can find *model based* estimates that consider the factor structure of the items. ω_h and ω_t are two such model based estimates and are easy to find in R.

```
                          ┌─ R code ─┐
> library(psych)     #make the psych package active
> library(psychTools) #make psychTools active
> om <- omega(Thurstone)  #do the analysis
> om  #show it
```

```
Omega
Call: omegah(m = m, nfactors = nfactors, fm = fm, key = key, flip = flip,
    digits = digits, title = title, sl = sl, labels = labels,
    plot = plot, n.obs = n.obs, rotate = rotate, Phi = Phi, option = option,
    covar = covar)
Alpha:                 0.89
G.6:                   0.91
```

```
Omega Hierarchical:     0.74
Omega H asymptotic:     0.79
Omega Total             0.93


Schmid Leiman Factor loadings greater than  0.2
                     g   F1*   F2*   F3*   h2   u2   p2
Sentences          0.71  0.56              0.82 0.18 0.61
Vocabulary         0.73  0.55              0.84 0.16 0.63
Sent.Completion    0.68  0.52              0.74 0.26 0.63
First.Letters      0.65        0.56        0.73 0.27 0.57
Four.Letter.Words  0.62        0.49        0.63 0.37 0.61
Suffixes           0.56        0.41        0.50 0.50 0.63
Letter.Series      0.59              0.62  0.73 0.27 0.48
Pedigrees          0.58  0.24          0.34 0.51 0.49 0.66
Letter.Group       0.54              0.46  0.52 0.48 0.56


With eigenvalues of:
   g   F1*   F2*   F3*
3.58 0.96 0.74 0.72


general/max  3.73   max/min =    1.34
mean percent general =  0.6    with sd =  0.05 and cv of  0.09
Explained Common Variance of the general factor =  0.6


The degrees of freedom are 12  and the fit is  0.01


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


Compare this with the adequacy of just a general factor and no group factors
The degrees of freedom for just the general factor are 27  and the fit is  1.48


The root mean square of the residuals is  0.14
The df corrected root mean square of the residuals is  0.16


Measures of factor score adequacy
                                               g   F1*   F2*   F3*
Correlation of scores with factors            0.86 0.73 0.72 0.75
Multiple R square of scores with factors      0.74 0.54 0.51 0.57
Minimum correlation of factor score estimates 0.49 0.07 0.03 0.13
```

```
 Total, General and Subset omega for each subset
                                                 g  F1*  F2*  F3*
Omega total for total scores and subscales     0.93 0.92 0.83 0.79
Omega general for total scores and subscales   0.74 0.58 0.50 0.47
Omega group for total scores and subscales     0.16 0.34 0.32 0.32

pdf
  2
```

Omega with Schmid Leiman Transformation

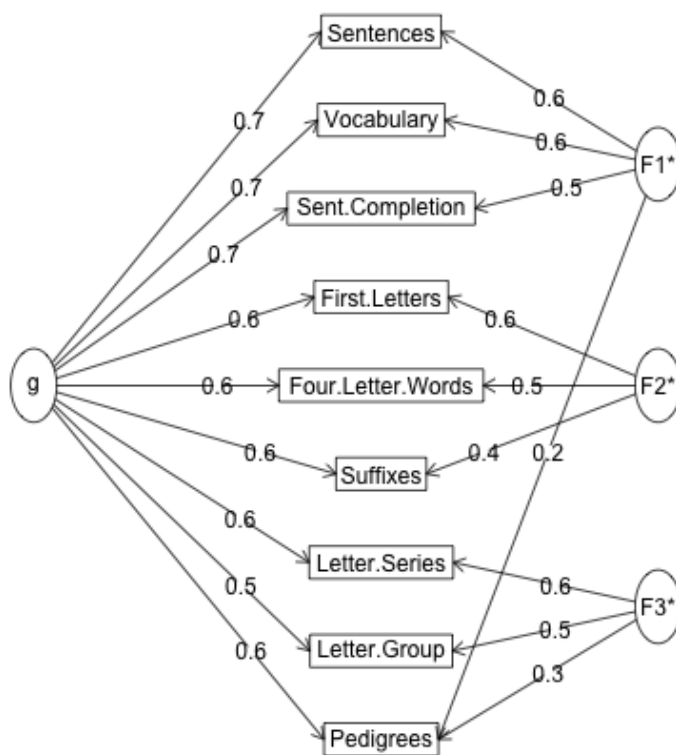Figure 1: ω_h is a reliability estimate of the general factor of a set of variables. It is based upon the correlation of lower order factors. It may be found in R by using the `omega` function which is part of the *psych* package. The figure shows a solution for the `Thurstone` 9 variable data set. Compare this to the solution using the `omegaDirect` function from Waller (2017) (Figure 3)

6

To use R obviously requires installing R on your computer. This is very easy to do (see section 1.2.1) and needs to be done once. (The following sections are elaborated in the "getting startedHow To" . If you need more help in installing R see the longer version.)

The power of R is in the supplemental *packages*. There are at least 16,000 packages that have been contributed to the R project. To do any of the analyses discussed in these "How To's", you will need to install the package *psych* (Revelle, 2020). To do factor analyses or principal component analyses you will also need the *GPArotation* (Bernaards and Jennrich, 2005) package. With these two packages, you will be be able to find ω_h using Exploratory Factor Analysis. If you want to find to estimate ω_h using Confirmatory Factor Analysis, you will also need to add the *lavaan* (Rosseel, 2012) package. To use *psych* to create simulated data sets, you also need the *mnormt* (Azzalini and Genz, 2016) package. For a more complete installation of a number of psychometric packages, you can install and activate a package (*ctv*) that installs a large set of psychometrically relevant packages. As is true for R, you will need to install packages just once.

1.2   Install R for the first time

1. Download from R Cran (http://cran.r-project.org/)

2. Install R (current version is 4.0.2)

3. Start R.

   Note that the R prompt > starts off every line. This is R's way of indicating that it wants input. In addition, note that almost all commands start and finish with parentheses.

4. Add useful packages (just need to do this once) (see section 1.2.2)

   ```R code
   install.packages("psych",dependencies=TRUE)   #the minimum requirement or
   install.packages(c("psych","GPArotation"),dependencies=TRUE)   #required for factor analysis
   ```

   (a) or if you want to do CFA

   ```R code
   install.packages(c("psych","lavaan"), dependencies=TRUE)
   ```

   (b) or if you want to install the psychometric task views

   ```R code
   install.packages("ctv")         #this downloads the task view package
   library(ctv)                    #this activates the ctv package
   install.views("Psychometrics")  #among others
   ```

5. Take a 5 minute break while the packages are loaded.

6. Activate the package(s) you want to use (e.g., *psych*)

```R code
library(psych)     #Only need to make psych active once a session
```

psych will automatically activate the other packages it needs, as long as they are installed. Note that *psych* is updated roughly quarterly, the current version is 2.0.8 Patches and improvements to *psych* (the bleeding edge version) are available from the repository at the personality-project web server and may be installed from there:

```R code
install.packages("psych", repos = "https://personality-project.org/r", type="source")
```

7. Use R

1.2.1 Install R

Go to the Comprehensive R Archive Network (CRAN) at `http://cran.r-project.org`: Choose your operating system and then download and install the appropriate version

Download and install the appropriate version – Mac, PC or Unix/Linux

Once you have installed R you probably, and particularly if you have a PC, will want to download and install the R Studio program. It is a very nice interface for PCs and Macs that combines four windows into one screen.

1.2.2 Install relevant packages

Once R is installed on your machine, you still need to install a few relevant "packages". Packages are what make R so powerful, for they are special sets of functions that are designed for one particular application. In the case of the *psych* package, this is an application for doing the kind of basic data analysis and psychometric analysis that psychologists and many others find particularly useful. *psych* may be thought of a "Swiss Army Knife" for psychological statistics. While not the best tool for a particular job, it is a useful tool for many jobs.

You may either install the minimum set of packages necessary to do the analysis using an Exploratory Factor Analysis (EFA) approach (recommended) or a few more packages to do both an EFA and a CFA approach. It is also possible to add many psychometrically relevant packages all at once by using the "task views" approach. A particularly powerful package is the *lavaan* (Rosseel, 2012) package for doing structural equation modeling. Another useful one is the *sem* pacakge (Fox et al., 2013).

Install the minimum set   This may be done by typing into the console or using menu options (e.g., the Package Installer underneath the Packages and Data menu).

```R code
install.packages(c("psych", "psychTools"), dependencies = TRUE)
```

Install a few more packages   If you want some more functionality for some of the more advanced statistical procedures (e.g., `omegaSem`) you will need to install a few more packages (e.g., *lavaan*.

```R code
install.packages(c("psych","GPArotation","lavaan"),dependencies=TRUE)
```

Install a "task view" to get lots of packages   If you know that there are a number of packages that you want to use, it is possible they are listed as a "task view". For instance, about 50 packages will be installed at once if you install the "psychometrics" task view. You can Install all the psychometric packages from the "psychometrics" task view by first installing a package ("ctv") that in turn installs many different task views. To see the list of possible task views, go to https://cran.r-project.org/web/views/.

```R code
install.packages("ctv") }       #this downloads the task view package
library(ctv)                            #this activates the ctv package
install.views("Psychometrics")     #one of the many Taskviews
```

Take a 5 minute break because you will be installing about 50 packages.

For the more adventurous users   The *psych* pacakge is under (sporadic) development with a new release issued to CRAN roughly every 4-6 months. The experimental, development version (prerelease) is always available at the Personality-Project web site and may be installed for Macs or PCs directly:

```R code
install.packages("psych", repos= "https://personality-project.org/r", type ="source")
```

This development version will have fixed any bugs reported since the last release and will have various new features that are being tested before release to CRAN. After installation, it is necessary to restart R to make the new version active.

Make the *psych* package active.   You are almost ready. But first, to use most of the following examples you need to make the *psych* and *psychTools* packages active. You only need to do this once per session.

```R code
library(psych)           #to do the analyses described here
library(psychTools)  #for some useful additions such as read.file
```

9

2 Reading in the data for analysis

2.1 Find a file and read from it

There are of course many ways to enter data into R. Reading from a local file using `read.file` is perhaps the most preferred. This will read in most of the standard file types (.csv, .sav, .txt, etc). `read.file` combines the `file.choose` and `read.table` functions:

```
R code
my.data <- read.file()   #note the open and closing parentheses
```

`read.file` opens a search window on your system just like any open file command does. `read.file` assumes that the first row of your table has labels for each column. If this is not true, specify names=FALSE, e.g.,

```
R code
my.data <- read.file(names = FALSE)
```

If you want to read a remote file, specify the file name and then `read.file`

```
R code
datafilename <- "http://personality-project.org/r/datasets/finkel.sav"
new.data <- read.file(datafilename)   #the data has labels
```

2.2 Or: copy the data from another program using the copy and paste commands of your operating system

However, many users will enter their data in a text editor or spreadsheet program and then want to copy and paste into R. This may be done by using one of the `read.clipboard` set of functions .

`read.clipboard` is the base function for reading data from the clipboard.

`read.clipboard.csv` for reading text that is comma delimited.

`read.clipboard.tab` for reading text that is tab delimited (e.g., copied directly from an Excel file).

`read.clipboard.lower` for reading input of a lower triangular matrix with or without a diagonal. The resulting object is a square matrix.

`read.clipboard.upper` for reading input of an upper triangular matrix.

`read.clipboard.fwf` for reading in fixed width fields (some very old data sets)

For example, given a data set copied to the clipboard from a spreadsheet, just enter the command

```R code
my.data <- read.clipboard()
```

This will work if every data field has a value and even missing data are given some values (e.g., NA or -999). If the data were entered in a spreadsheet and the missing values were just empty cells, then the data should be read in as a tab delimited or by using the `read.clipboard.tab` function.

```R code
my.data <- read.clipboard(sep="\t")    #define the tab option, or
my.tab.data <- read.clipboard.tab()    #just use the alternative function
```

For the case of data in fixed width fields (some old data sets tend to have this format), copy to the clipboard and then specify the width of each field (in the example below, the first variable is 5 columns, the second is 2 columns, the next 5 are 1 column the last 4 are 3 columns).

```R code
my.data <- read.clipboard.fwf(widths=c(5,2,rep(1,5),rep(3,4))
```

2.3   Or: import from an SPSS or SAS file

To read data from an SPSS, SAS, or Systat file, you can probably just use the `read.file` function. `read.file` examines the suffix of the data file and if it is .sav (from SPSS) or .xpt (from SAS) will attempt to read given various default options. However, if that does not work, use the *foreign* package. This should come with Base R but still need to be loaded using the `library` command.

`read.spss` reads a file stored by the SPSS save or export commands.

```
read.spss(file, use.value.labels = TRUE, to.data.frame = FALSE,
          max.value.labels = Inf, trim.factor.names = FALSE,
          trim_values = TRUE, reencode = NA, use.missings = to.data.frame)
```

The `read.spss` function has many parameters that need to be set. In the example, I have used the parameters that I think are most useful.

file Character string: the name of the file or URL to read.

use.value.labels Convert variables with value labels into R factors with those levels?

to.data.frame return a data frame? Defaults to FALSE, probably should be TRUE in most cases.

max.value.labels Only variables with value labels and at most this many unique values will be converted to factors if use.value.labels $= TRUE$.

trim.factor.names Logical: trim trailing spaces from factor levels?

trim_values logical: should values and value labels have trailing spaces ignored when matching for use.value.labels $= TRUE$?

use.missings logical: should information on user-defined missing values be used to set the corresponding values to NA?

The following is an example of reading from a remote SPSS file and then describing the data set to make sure that it looks ok (with thanks to Eli Finkel).

```R code
datafilename <- "http://personality-project.org/r/datasets/finkel.sav"
eli <-read.file(datafilename)


describe(eli,skew=FALSE)
```

	var	n	mean	sd	median	trimmed	mad	min	max	range	se
USER*	1	69	35.00	20.06	35	35.00	25.20	1	69	68	2.42
HAPPY	2	69	5.71	1.04	6	5.82	0.00	2	7	5	0.13
SOULMATE	3	69	5.09	1.80	5	5.32	1.48	1	7	6	0.22
ENJOYDEX	4	68	6.47	1.01	7	6.70	0.00	2	7	5	0.12
UPSET	5	69	0.41	0.49	0	0.39	0.00	0	1	1	0.06

3 Some simple descriptive statistics before you start

Although you probably want to jump right in and find ω, you should first make sure that your data are reasonable. Use the `describe` function to get some basic descriptive statistics. This next example takes advantage of a built in data set.

```R code
my.data <- sat.act   #built in example  -- replace with your data
describe(my.data)
```

	var	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtos
gender	1	700	1.65	0.48	2	1.68	0.00	1	2	1	-0.61	-1.
education	2	700	3.16	1.43	3	3.31	1.48	0	5	5	-0.68	-0.
age	3	700	25.59	9.50	22	23.86	5.93	13	65	52	1.64	2.

```
ACT          4 700  28.55   4.82    29  28.84   4.45   3  36   33 -0.66     0.
SATV         5 700 612.23 112.90   620 619.45 118.61 200 800  600 -0.64     0.
SATQ         6 687 610.22 115.64   620 617.25 118.61 200 800  600 -0.59    -0.
```

There are, of course, all kinds of things you could do with your data at this point, but read about them in the introductory vignette and more advanced vignette for the *psych* package,

4   Using the `omega` function to find ω

Two alternative estimates of reliability that take into account the hierarchical structure of the inventory are McDonald's ω_h and ω_t (McDonald, 1999; Revelle and Zinbarg, 2009). These may be found using the `omega` function for an exploratory analysis (See Figure 1) or `omegaSem` for a confirmatory analysis using the *sem* based upon the exploratory solution from `omega`.

4.1   Background on the ω statistics

McDonald (1999) has proposed coefficient omega (hierarchical) (ω_h) as an estimate of the general factor saturation of a test. Zinbarg et al. (2005) compare McDonald's ω_h to Cronbach's α and Revelle's β. They conclude that ω_h is the best estimate. (See also Zinbarg et al. (2006) and Revelle and Zinbarg (2009)).

One way to find ω_h is to do a factor analysis of the original data set, rotate the factors obliquely, factor that correlation matrix, do a Schmid-Leiman (`schmid`) transformation to find general factor loadings, and then find ω_h.

ω_h differs slightly as a function of how the factors are estimated. Three options are available, the default will do a minimum residual factor analysis, fm="pa" does a principal axes factor analysis (`factor.pa`), and fm="mle" provides a maximum likelihood solution.

For ability items, it is typically the case that all items will have positive loadings on the general factor. However, for non-cognitive items it is frequently the case that some items are to be scored positively, and some negatively. Although probably better to specify which directions the items are to be scored by specifying a key vector, if flip =TRUE (the default), items will be reversed so that they have positive loadings on the general factor. The keys are reported so that scores can be found using the `score.items` function. Arbitrarily reversing items this way can overestimate the general factor. (See the example with a simulated circumplex).

The `omega` function uses exploratory factor analysis to estimate the ω_h coefficient. It is important to remember that "A recommendation that should be heeded, regardless of the method chosen to estimate ω_h, is to always examine the pattern of the estimated general

13

factor loadings prior to estimating ω_h. Such an examination constitutes an informal test of the assumption that there is a latent variable common to all of the scale's indicators that can be conducted even in the context of EFA. If the loadings were salient for only a relatively small subset of the indicators, this would suggest that there is no true general factor underlying the covariance matrix. Just such an informal assumption test would have afforded a great deal of protection against the possibility of misinterpreting the misleading ω_h estimates occasionally produced in the simulations reported here." (Zinbarg et al., 2006, p 137).

Although ω_h is uniquely defined only for cases where 3 or more subfactors are extracted, it is sometimes desired to have a two factor solution. By default this is done by forcing the `schmid` extraction to treat the two subfactors as having equal loadings.

There are three possible options for this condition: setting the general factor loadings between the two lower order factors to be "equal" which will be the $\sqrt{r_{ab}}$ where r_{ab} is the oblique correlation between the factors) or to "first" or "second" in which case the general factor is equated with either the first or second group factor. A message is issued suggesting that the model is not really well defined. This solution discussed in Zinbarg et al., 2007. To do this in omega, add the option="first" or option="second" to the call.

Although obviously not meaningful for a 1 factor solution, it is of course possible to find the sum of the loadings on the first (and only) factor, square them, and compare them to the overall matrix variance. This is done, with appropriate complaints.

In addition to ω_h, another of McDonald's coefficients is ω_t. This is an estimate of the total reliability of a test.

McDonald's ω_t, which is similar to Guttman's λ_6, (see `guttman`) uses the estimates of uniqueness u^2 from factor analysis to find e_j^2. This is based on a decomposition of the variance of a test score, V_x into four parts: that due to a general factor, g, that due to a set of group factors, f, (factors common to some but not all of the items), specific factors, s unique to each item, and e, random error. (Because specific variance can not be distinguished from random error unless the test is given at least twice, some combine these both into error).

Letting $x = cg + Af + Ds + e$ then the communality of item$_j$, based upon general as well as group factors, $h_j^2 = c_j^2 + \sum f_{ij}^2$ and the unique variance for the item $u_j^2 = \sigma_j^2(1 - h_j^2)$ may be used to estimate the test reliability. That is, if h_j^2 is the communality of item$_j$, based upon general as well as group factors, then for standardized items, $e_j^2 = 1 - h_j^2$ and

$$\omega_t = \frac{1cc'1 + 1AA'1'}{V_x} = 1 - \frac{\sum(1 - h_j^2)}{V_x} = 1 - \frac{\sum u^2}{V_x}$$

Because $h_j^2 \geq r_{smc}^2$, $\omega_t \geq \lambda_6$.

It is important to distinguish here between the two ω coefficients of McDonald, 1978 and Equation 6.20a of McDonald, 1999, ω_t and ω_h. While the former is based upon the sum of squared loadings on all the factors, the latter is based upon the sum of the squared loadings on the general factor.

$$\omega_h = \frac{1cc'1}{V_x}$$

Another estimate reported is the omega for an infinite length test with a structure similar to the observed test. This is found by

$$\omega_{\mathrm{inf}} = \frac{1cc'1}{1cc'1 + 1AA'1'}$$

It can be shown In the case of simulated variables, that the amount of variance attributable to a general factor (ω_h) is quite large, and the reliability of the set of items is somewhat greater than that estimated by α or λ_6.

4.2 Yet another alternative: Coefficient β

β, an alternative to ω_h, is defined as the worst split half reliability (Revelle, 1979). It can be estimated by using iclust (Item Cluster analysis: a hierarchical clustering algorithm). For a very complimentary review of why the iclust algorithm is useful in scale construction, see Cooksey and Soutar (2006). For a discussion of how use iclust see the factor analysis vignette.

4.3 Using the omega function

This is R. Just call it. For the next example, we find ω for a data set from Thurstone. To find it for your data, replace Thurstone with my.data.

R code

```
> omega(Thurstone)
```

```
Omega
Call: omegah(m = m, nfactors = nfactors, fm = fm, key = key, flip = flip,
    digits = digits, title = title, sl = sl, labels = labels,
    plot = plot, n.obs = n.obs, rotate = rotate, Phi = Phi, option = option,
    covar = covar)
Alpha:                 0.89
G.6:                   0.91
Omega Hierarchical:    0.74
Omega H asymptotic:    0.79
Omega Total            0.93
```

```
Schmid Leiman Factor loadings greater than  0.2
                          g   F1*   F2*   F3*   h2   u2   p2
Sentences            0.71  0.56             0.82 0.18 0.61
Vocabulary           0.73  0.55             0.84 0.16 0.63
Sent.Completion      0.68  0.52             0.74 0.26 0.63
First.Letters        0.65        0.56       0.73 0.27 0.57
Four.Letter.Words 0.62           0.49       0.63 0.37 0.61
Suffixes             0.56        0.41       0.50 0.50 0.63
Letter.Series        0.59              0.62 0.73 0.27 0.48
Pedigrees            0.58  0.24        0.34 0.51 0.49 0.66
Letter.Group         0.54              0.46 0.52 0.48 0.56


With eigenvalues of:
   g  F1*  F2*  F3*
3.58 0.96 0.74 0.72


general/max  3.73   max/min =   1.34
mean percent general =  0.6    with sd =  0.05 and cv of  0.09
Explained Common Variance of the general factor =  0.6


The degrees of freedom are 12  and the fit is  0.01


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


Compare this with the adequacy of just a general factor and no group factors
The degrees of freedom for just the general factor are 27  and the fit is  1.48


The root mean square of the residuals is  0.14
The df corrected root mean square of the residuals is  0.16


Measures of factor score adequacy
                                             g   F1*  F2*  F3*
Correlation of scores with factors          0.86 0.73 0.72 0.75
Multiple R square of scores with factors    0.74 0.54 0.51 0.57
Minimum correlation of factor score estimates 0.49 0.07 0.03 0.13


 Total, General and Subset omega for each subset
                                             g   F1*  F2*  F3*
Omega total for total scores and subscales  0.93 0.92 0.83 0.79
```

```
Omega general for total scores and subscales  0.74 0.58 0.50 0.47
Omega group for total scores and subscales    0.16 0.34 0.32 0.32
```

4.4 Find three measures of reliability: ω_h, α, and ω_t

In a review of various measures of reliability, Revelle and Condon (2019) suggest that one should routinely report 3 estimates of internal consistency (ω_h, α, and ω_t). As an example, they use 10 items to measure anxiety taken from the state anxiety data set (`sai` in the *psychTools* package. First examine the descriptive statistics and then find and summarize the omega for these data. By inspection of the correlation matrix, it seems as if there are two group factors (tension and calmness) as well as an overall general factor of anxiety. We use a two factor solution to better represent the results (Figure 2).

```
R code
> anxiety <- sai[c("anxious", "jittery", "nervous" ,"tense", "upset","at.ease" ,  "calm" ,
+    "confident", "content","relaxed")]
> describe(anxiety)
```

```
          vars      n mean   sd median trimmed  mad min max range   skew kurtosis
anxious      1 5317 1.69 0.84      1    1.57 0.00   1   4     3   1.03     0.25
jittery      2 5291 1.51 0.81      1    1.34 0.00   1   4     3   1.55     1.61
nervous      3 5297 1.43 0.71      1    1.28 0.00   1   4     3   1.64     2.15
tense        4 5338 1.65 0.84      1    1.51 0.00   1   4     3   1.13     0.41
upset        5 5330 1.37 0.70      1    1.21 0.00   1   4     3   1.98     3.45
at.ease      6 5334 2.60 0.89      3    2.62 1.48   1   4     3   0.03    -0.78
calm         7 5345 2.73 0.89      3    2.75 1.48   1   4     3  -0.01    -0.94
confident    8 5302 2.68 0.91      3    2.73 1.48   1   4     3  -0.14    -0.80
content      9 5275 2.48 0.93      2    2.47 1.48   1   4     3   0.06    -0.85
relaxed     10 5280 2.48 0.90      2    2.48 1.48   1   4     3   0.10    -0.75
```

```
R code
> lowerCor(anxiety)
```

```
          anxis jttry nervs tense upset at.es calm   cnfdn cntnt relxd
anxious    1.00
jittery    0.58  1.00
nervous    0.58  0.60  1.00
tense      0.59  0.56  0.63  1.00
upset      0.33  0.22  0.39  0.47  1.00
at.ease   -0.33 -0.34 -0.37 -0.47 -0.36  1.00
calm      -0.37 -0.40 -0.38 -0.49 -0.32  0.70  1.00
confident -0.06 -0.05 -0.14 -0.17 -0.26  0.47  0.36  1.00
content   -0.18 -0.15 -0.22 -0.32 -0.37  0.60  0.52  0.56  1.00
relaxed   -0.34 -0.39 -0.37 -0.48 -0.33  0.69  0.70  0.41  0.59  1.00
```

```R code
> om <- omega(anxiety,2)  #specify a two factor solution
> summary(om)    #summarize the output
```

```
Omega
omega(m = anxiety, nfactors = 2)
Alpha:                    0.87
G.6:                      0.89
Omega Hierarchical:       0.57
Omega H asymptotic:       0.62
Omega Total               0.91

With eigenvalues of:
  g F1* F2*
2.7 1.5 1.4
The degrees of freedom for the model is 26  and the fit was  0.24
The number of observations was  5378  with Chi Square =  1298.24  with prob <  (

The root mean square of the residuals is  0.04
The df corrected root mean square of the residuals is  0.06

RMSEA and the  0.9 confidence intervals are  0.095 0.091 0.1
BIC =  1074.89Explained Common Variance of the general factor =  0.48

 Total, General and Subset omega for each subset
                                                g  F1*  F2*
Omega total for total scores and subscales    0.91 0.86 0.85
Omega general for total scores and subscales  0.57 0.44 0.40
Omega group for total scores and subscales    0.29 0.42 0.45

pdf
  2
```

4.5   Estimating ω_h using a direct Schmid-Leiman transformation

The `omegaDirect` function uses Niels Waller's algorithm for finding a g factor directly without extracting a higher order model (Waller, 2017). This has the advantage that it will work cleanly for data with just 2 group factors. Unfortunately, it will produce non-zero estimates for omega even if there is no general factor.

```R code
> om <- omegaDirect(Thurstone)
```

Omega analysis of two factors of anxiety
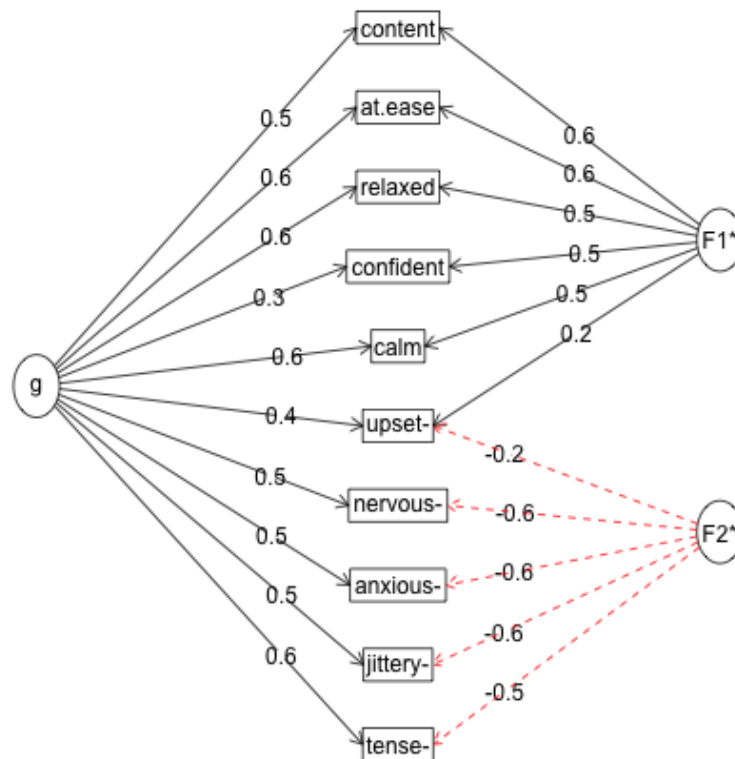
Figure 2: An `omega` solution for 10 anxiety items with two group factors. See Revelle and Condon (2019) for more measures of reliability for this data set.

```
> om
```

Call: omegaDirect(m = Thurstone)

Omega from direct Schmid Leiman = 0.7

Call: omegaDirect(m = Thurstone)
Standardized loadings (pattern matrix) based upon correlation matrix
                     g   F1*   F2*   F3*   h2   u2
Sentences   0.68 0.59 0.01   0.08 0.82 0.18
Vocabulary   0.70 0.58 0.08   0.03 0.84 0.16
Sent.Completion   0.66 0.55 0.05   0.05 0.74 0.26
First.Letters   0.62 0.02 0.58   0.03 0.73 0.27
Four.Letter.Words 0.60 0.00 0.51   0.10 0.63 0.37
Suffixes   0.54 0.13 0.43 -0.03 0.50 0.50
Letter.Series   0.59 -0.01 -0.02   0.62 0.73 0.27
Pedigrees   0.57 0.23 -0.02   0.36 0.51 0.49
Letter.Group   0.54 -0.06 0.14   0.46 0.52 0.48

                          g   F1* F2* F3*
SS loadings   3.39 1.06 0.81 0.74
Proportion Var   0.38 0.12 0.09 0.08
Cumulative Var   0.38 0.49 0.58 0.67
Proportion Explained   0.56 0.18 0.14 0.12
Cumulative Proportion 0.56 0.74 0.88 1.00

With eigenvalues of:
   g F1* F2* F3*
3.39 1.06 0.81 0.74
The degrees of freedom for the model is 12 and the fit was 0.01

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

 Total, General and Subset omega for each subset
                                              g   F1* F2* F3*
Omega total for total scores and subscales   0.93 0.92 0.82 0.79
Omega general for total scores and subscales 0.70 0.53 0.47 0.46
Omega group for total scores and subscales   0.17 0.38 0.35 0.33

pdf

Direct Schmid Leihman solution
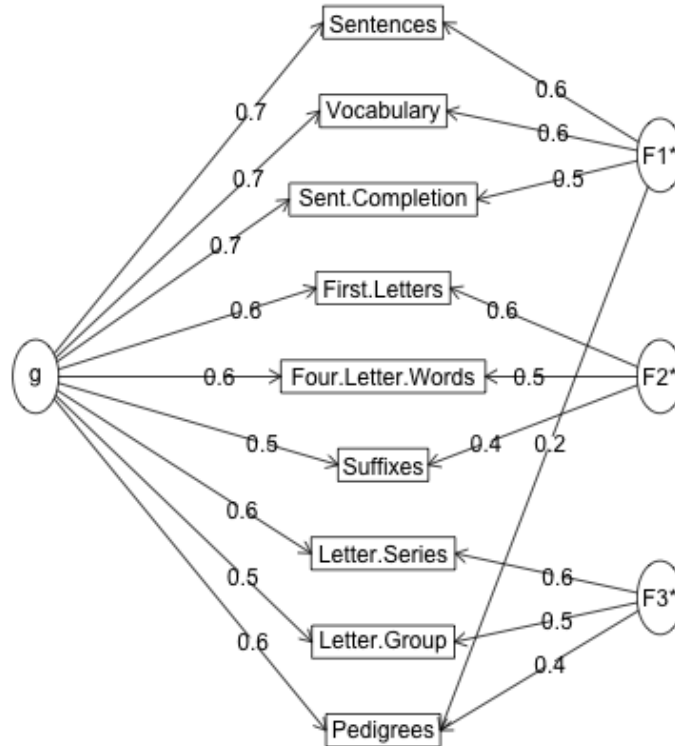
Figure 3: The Direct Schmid Leiman solution is taken from an algorithm by Waller (2017). Compare this solution to Figure 1.

4.6   Estimating ω_h using Confirmatory Factor Analysis

The `omegaSem` function will do an exploratory analysis and then take the highest loading items on each factor and do a confirmatory factor analysis using the *lavaan* package. These results can produce slightly different estimates of ω_h, primarily because cross loadings are modeled as part of the general factor.

We use a classic data set from Holzinger and Swineford, some of the tests of which are included in the *lavaan* package. This analysis allows us to examine the hierarchical structure of these ability tests. The data are taken from the `holzinger.swineford` data set in the

psychTools package.

```R code
> om <- omega(holzinger.swineford[8:31],4)  #the exploratory solution
```

```
Omega
Call: omegah(m = m, nfactors = nfactors, fm = fm, key = key, flip = flip,
    digits = digits, title = title, sl = sl, labels = labels,
    plot = plot, n.obs = n.obs, rotate = rotate, Phi = Phi, option = option,
    covar = covar)
Alpha:                   0.89
G.6:                     0.92
Omega Hierarchical:      0.58
Omega H asymptotic:      0.63
Omega Total              0.92


Schmid Leiman Factor loadings greater than  0.2
                g    F1*   F2*   F3*   F4*   h2   u2   p2
t01_visperc   0.49        0.47              0.47 0.53 0.50
t02_cubes     0.29        0.42              0.28 0.72 0.30
t03_frmbord   0.29        0.35              0.22 0.78 0.39
t04_lozenges  0.40        0.52              0.44 0.56 0.37
t05_geninfo   0.44  0.70                    0.69 0.31 0.28
t06_paracomp  0.49  0.66                    0.67 0.33 0.36
t07_sentcomp  0.44  0.76                    0.78 0.22 0.25
t08_wordclas  0.47  0.57                    0.55 0.45 0.41
t09_wordmean  0.51  0.67                    0.71 0.29 0.36
t10_addition  0.29              0.71        0.62 0.38 0.14
t11_code      0.45              0.47        0.47 0.53 0.43
t12_countdot  0.34              0.57        0.46 0.54 0.26
t13_sccaps    0.42        0.29  0.44        0.43 0.57 0.41
t14_wordrecg  0.32                    0.57  0.43 0.57 0.24
t15_numbrecg  0.26                    0.51  0.35 0.65 0.19
t16_figrrecg  0.45        0.25        0.38  0.41 0.59 0.50
t17_objnumb   0.30              0.24  0.46  0.39 0.61 0.24
t18_numbfig   0.34                    0.36  0.27 0.73 0.42
t19_figword   0.36                    0.27  0.23 0.77 0.55
t20_deduction 0.45        0.32              0.38 0.62 0.53
t21_numbpuzz  0.50        0.29  0.25        0.41 0.59 0.60
t22_probreas  0.49  0.34  0.27              0.43 0.57 0.55
t23_series    0.56  0.24  0.38              0.52 0.48 0.59
```

```
t24_woody      0.49  0.30           0.25        0.42 0.58 0.57
```

With eigenvalues of:
```
  g F1* F2* F3* F4*
4.2 2.7 1.4 1.5 1.3
```

```
general/max  1.57   max/min =   2.12
```
mean percent general = 0.39   with sd = 0.14 and cv of 0.34
Explained Common Variance of the general factor = 0.38

The degrees of freedom are 186 and the fit is 0.92
The number of observations was 301 with Chi Square = 264.96 with prob < 0.0
The root mean square of the residuals is 0.03
The df corrected root mean square of the residuals is 0.04
RMSEA index = 0.037 and the 10 % confidence intervals are 0.027 0.048
BIC = -796.56

Compare this with the adequacy of just a general factor and no group factors
The degrees of freedom for just the general factor are 252 and the fit is 4.67
The number of observations was 301 with Chi Square = 1357.46 with prob < 1e
The root mean square of the residuals is 0.15
The df corrected root mean square of the residuals is 0.15

RMSEA index = 0.121 and the 10 % confidence intervals are 0.115 0.127
BIC = -80.73

Measures of factor score adequacy
```
                                              g  F1*  F2*  F3*  F4*
Correlation of scores with factors         0.77 0.86 0.74 0.82 0.76
Multiple R square of scores with factors   0.59 0.73 0.55 0.68 0.58
Minimum correlation of factor score estimates 0.17 0.47 0.09 0.36 0.17
```

 Total, General and Subset omega for each subset
```
                                              g  F1*  F2*  F3*  F4*
Omega total for total scores and subscales   0.92 0.90 0.77 0.77 0.71
Omega general for total scores and subscales 0.58 0.37 0.41 0.25 0.28
Omega group for total scores and subscales   0.21 0.53 0.35 0.52 0.44
```

```
┌─ R code ─────────────────────────────────────────────────────────┐
│ > omegaSem(holzinger.swineford[8:31],4) #the confirmatory solution │
└───────────────────────────────────────────────────────────────────┘
```

Call: omegaSem(m = holzinger.swineford[8:31], nfactors = 4)

```
Omega
Call: omegah(m = m, nfactors = nfactors, fm = fm, key = key, flip = flip,
    digits = digits, title = title, sl = sl, labels = labels,
    plot = plot, n.obs = n.obs, rotate = rotate, Phi = Phi, option = option,
    covar = covar)
Alpha:                    0.89
G.6:                      0.92
Omega Hierarchical:       0.58
Omega H asymptotic:       0.63
Omega Total               0.92
```

Schmid Leiman Factor loadings greater than 0.2

	g	F1*	F2*	F3*	F4*	h2	u2	p2
t01_visperc	0.49		0.47			0.47	0.53	0.50
t02_cubes	0.29		0.42			0.28	0.72	0.30
t03_frmbord	0.29		0.35			0.22	0.78	0.39
t04_lozenges	0.40		0.52			0.44	0.56	0.37
t05_geninfo	0.44	0.70				0.69	0.31	0.28
t06_paracomp	0.49	0.66				0.67	0.33	0.36
t07_sentcomp	0.44	0.76				0.78	0.22	0.25
t08_wordclas	0.47	0.57				0.55	0.45	0.41
t09_wordmean	0.51	0.67				0.71	0.29	0.36
t10_addition	0.29			0.71		0.62	0.38	0.14
t11_code	0.45			0.47		0.47	0.53	0.43
t12_countdot	0.34			0.57		0.46	0.54	0.26
t13_sccaps	0.42		0.29	0.44		0.43	0.57	0.41
t14_wordrecg	0.32				0.57	0.43	0.57	0.24
t15_numbrecg	0.26				0.51	0.35	0.65	0.19
t16_figrrecg	0.45		0.25		0.38	0.41	0.59	0.50
t17_objnumb	0.30			0.24	0.46	0.39	0.61	0.24
t18_numbfig	0.34				0.36	0.27	0.73	0.42
t19_figword	0.36				0.27	0.23	0.77	0.55
t20_deduction	0.45		0.32			0.38	0.62	0.53
t21_numbpuzz	0.50		0.29	0.25		0.41	0.59	0.60
t22_probreas	0.49	0.34	0.27			0.43	0.57	0.55
t23_series	0.56	0.24	0.38			0.52	0.48	0.59
t24_woody	0.49	0.30		0.25		0.42	0.58	0.57

```
With eigenvalues of:
  g F1* F2* F3* F4*
4.2 2.7 1.4 1.5 1.3
```

```
general/max  1.57    max/min =    2.12
mean percent general =  0.39    with sd =  0.14 and cv of  0.34
Explained Common Variance of the general factor =  0.38


The degrees of freedom are 186  and the fit is  0.92
The number of observations was  301  with Chi Square =  264.96  with prob <  0.0
The root mean square of the residuals is  0.03
The df corrected root mean square of the residuals is  0.04
RMSEA index =  0.037  and the 10 % confidence intervals are  0.027 0.048
BIC =  -796.56


Compare this with the adequacy of just a general factor and no group factors
The degrees of freedom for just the general factor are 252  and the fit is  4.67
The number of observations was  301  with Chi Square =  1357.46  with prob <  1e
The root mean square of the residuals is  0.15
The df corrected root mean square of the residuals is  0.15


RMSEA index =  0.121  and the 10 % confidence intervals are  0.115 0.127
BIC =  -80.73


Measures of factor score adequacy
                                                 g  F1* F2* F3* F4*
Correlation of scores with factors            0.77 0.86 0.74 0.82 0.76
Multiple R square of scores with factors      0.59 0.73 0.55 0.68 0.58
Minimum correlation of factor score estimates 0.17 0.47 0.09 0.36 0.17


 Total, General and Subset omega for each subset
                                                 g  F1* F2* F3* F4*
Omega total for total scores and subscales    0.92 0.90 0.77 0.77 0.71
Omega general for total scores and subscales  0.58 0.37 0.41 0.25 0.28
Omega group for total scores and subscales    0.21 0.53 0.35 0.52 0.44


 The following analyses were done using the  lavaan  package

 Omega Hierarchical from a confirmatory model using sem =  0.77
 Omega Total  from a confirmatory model using sem =  0.92
With loadings of
                 g  F1* F2* F3* F4*   h2   u2   p2
t01_visperc   0.56     0.40           0.47 0.53 0.67
t02_cubes     0.33     0.37           0.24 0.76 0.45
```

```
t03_frmbord  0.31      0.38                0.24 0.76 0.40
t04_lozenges 0.42      0.51                0.44 0.56 0.40
t05_geninfo  0.50 0.67                     0.70 0.30 0.36
t06_paracomp 0.56 0.60                     0.67 0.33 0.47
t07_sentcomp 0.50 0.73                     0.77 0.23 0.32
t08_wordclas 0.55 0.50                     0.55 0.45 0.55
t09_wordmean 0.59 0.61                     0.72 0.28 0.48
t10_addition 0.30           0.60          0.45 0.55 0.20
t11_code     0.49           0.46          0.45 0.55 0.53
t12_countdot 0.36           0.59          0.47 0.53 0.28
t13_sccaps   0.46           0.43          0.40 0.60 0.53
t14_wordrecg 0.32                0.55 0.40 0.60 0.26
t15_numbrecg 0.23                0.57 0.37 0.63 0.14
t16_figrrecg 0.53                0.32 0.39 0.61 0.72
t17_objnumb  0.29                0.47 0.31 0.69 0.27
t18_numbfig  0.35                0.37 0.26 0.74 0.47
t19_figword  0.47                     0.25 0.75 0.88
t20_deduction 0.56                    0.35 0.65 0.90
t21_numbpuzz 0.64                     0.42 0.58 0.98
t22_probreas 0.64                     0.44 0.56 0.93
t23_series   0.72                     0.55 0.45 0.94
t24_woody    0.65                     0.43 0.57 0.98
```

With sum of squared loadings of:
```
   g  F1*  F2*  F3*  F4*
5.75 2.00 0.76 1.11 1.11
```

The degrees of freedom of the confirmatory model are 228 and the fit is 424.7
general/max 2.88   max/min =   2.62
mean percent general = 0.55   with sd = 0.27 and cv of 0.49
Explained Common Variance of the general factor = 0.54

Measures of factor score adequacy

	g	F1*	F2*	F3*	F4*
Correlation of scores with factors	0.92	0.95	0.77	0.91	0.82
Multiple R square of scores with factors	0.84	0.89	0.60	0.82	0.67
Minimum correlation of factor score estimates	0.68	0.79	0.19	0.64	0.34

 Total, General and Subset omega for each subset

	g	F1*	F2*	F3*	F4*
Omega total for total scores and subscales	0.92	0.91	0.79	0.76	0.72

```
Omega general for total scores and subscales  0.77 0.53 0.59 0.28 0.32
Omega group for total scores and subscales    0.16 0.38 0.20 0.47 0.40

To get the standard sem fit statistics, ask for summary on the fitted object
```

5   Simulating a hierarchical/higher order structure

There are several simulation functions in the *psych* package for creating structures with a general factor. One, `sim.hierarchical` creates lower level factors which are all correlated with a general factor. The default simulation has the parameters discussed by Jensen and Weng (1994). Another way to simulate a hierarchical structure is to simulate a bifactor model directly using the `sim.structure` function.

The Jensen and Weng (1994) model:

```R code
> jen <- sim.hierarchical()   #use the default values
> om <- omega(jen)
> om
```

```
Omega
Call: omegah(m = m, nfactors = nfactors, fm = fm, key = key, flip = flip,
    digits = digits, title = title, sl = sl, labels = labels,
    plot = plot, n.obs = n.obs, rotate = rotate, Phi = Phi, option = option,
    covar = covar)
Alpha:                 0.76
G.6:                   0.76
Omega Hierarchical:    0.69
Omega H asymptotic:    0.86
Omega Total            0.8

Schmid Leiman Factor loadings greater than  0.2
      g   F1*  F2*  F3*   h2   u2   p2
V1 0.72 0.35           0.64 0.36 0.81
V2 0.63 0.31           0.49 0.51 0.81
V3 0.54 0.26           0.36 0.64 0.81
V4 0.56      0.42      0.49 0.51 0.64
V5 0.48      0.36      0.36 0.64 0.64
V6 0.40      0.30      0.25 0.75 0.64
V7 0.42           0.43 0.36 0.64 0.49
V8 0.35           0.36 0.25 0.75 0.49
V9 0.28           0.29 0.16 0.84 0.49
```

```
With eigenvalues of:
   g  F1*  F2*  F3*
2.29 0.28 0.40 0.39

general/max  5.78   max/min =   1.4
mean percent general =  0.65    with sd =  0.14 and cv of  0.21
Explained Common Variance of the general factor =  0.68

The degrees of freedom are 12  and the fit is  0

The root mean square of the residuals is  0
The df corrected root mean square of the residuals is  0

Compare this with the adequacy of just a general factor and no group factors
The degrees of freedom for just the general factor are 27  and the fit is  0.18

The root mean square of the residuals is  0.06
The df corrected root mean square of the residuals is  0.07

Measures of factor score adequacy
                                                 g   F1*   F2*   F3*
Correlation of scores with factors              0.85  0.46  0.57  0.57
Multiple R square of scores with factors        0.73  0.21  0.32  0.32
Minimum correlation of factor score estimates   0.46 -0.57 -0.35 -0.35

 Total, General and Subset omega for each subset
                                                 g  F1*  F2*  F3*
Omega total for total scores and subscales      0.80 0.74 0.63 0.50
Omega general for total scores and subscales    0.69 0.60 0.40 0.25
Omega group for total scores and subscales      0.11 0.14 0.23 0.26
```

5.0.1 Simulate a bifactor model

Simulate a bifactor model and then compare two ways of finding the solution (normal omega and directOmega). We compare the solutions using the `fa.congruence` function.

R code

```
fx <- matrix(c(.7,.6,.5,.7,.6,.5,.8,.7,.6,
    .6,.6,.6,rep(0,9),c(.6,.5,.6),rep(0,9),.6,.6,.6),ncol=4)
 simx <-sim.structure(fx)
 lowerMat(simx$model)


om <- omega(simx$model)
dsl <- omegaDirect(simx$model)
```
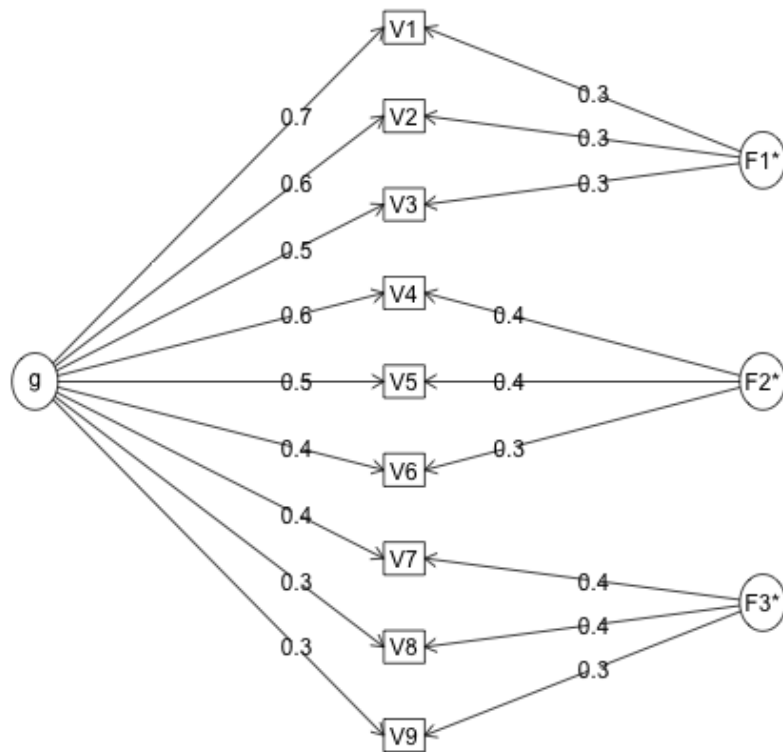
Omega with Schmid Leiman Transformation

Figure 4: An example of a hierarchical model from Jensen.

```
summary(om)
summary(dsl)
fa.congruence(list(om,dsl,fx))
```

> **R code**

```
>   lowerMat(simx$model)
```

```
    V1   V2   V3   V4   V5   V6   V7   V8   V9
V1 1.00
V2 0.78 1.00
V3 0.71 0.66 1.00
V4 0.49 0.42 0.35 1.00
V5 0.42 0.36 0.30 0.72 1.00
V6 0.35 0.30 0.25 0.71 0.60 1.00
V7 0.56 0.48 0.40 0.56 0.48 0.40 1.00
V8 0.49 0.42 0.35 0.49 0.42 0.35 0.92 1.00
V9 0.42 0.36 0.30 0.42 0.36 0.30 0.84 0.78 1.00
```

> **R code**

```
> summary(om)
```

```
Omega
omega(m = simx$model)
Alpha:                  0.9
G.6:                    0.93
Omega Hierarchical:     0.74
Omega H asymptotic:     0.78
Omega Total             0.95

With eigenvalues of:
   g  F1*  F2*  F3*
3.67 1.08 1.08 0.97
The degrees of freedom for the model is 12  and the fit was  0

The root mean square of the residuals is  0
The df corrected root mean square of the residuals is  0
Explained Common Variance of the general factor =  0.54

 Total, General and Subset omega for each subset
                                              g  F1*  F2*  F3*
Omega total for total scores and subscales    0.95 0.95 0.89 0.87
Omega general for total scores and subscales  0.74 0.55 0.45 0.46
Omega group for total scores and subscales    0.21 0.40 0.44 0.41
```

> **R code**

```
> summary(dsl)
```

```
Call: omegaDirect(m = simx$model)
Omega H direct:    0.71

With eigenvalues of:
   g  F1*  F2*  F3*
3.53 1.22 1.06 0.99
The degrees of freedom for the model is 12  and the fit was  0

The root mean square of the residuals is  0
The df corrected root mean square of the residuals is  0

 Total, General and Subset omega for each subset
                                              g  F1*  F2*  F3*
```

```
Omega total for total scores and subscales     0.95 0.95 0.89 0.87
Omega general for total scores and subscales   0.71 0.50 0.45 0.45
Omega group for total scores and subscales     0.22 0.45 0.43 0.42
```

R code

```
> fa.congruence(list(om,dsl,fx))
```

```
       g   F1*  F2*  F3*   h2    g   F1*  F2*  F3*        g    F1*  F2*  F3*
g    1.00 0.64 0.55 0.54 1.00 1.00 0.67 0.56 0.57 1.00 0.54 0.54 0.63
F1*  0.64 1.00 0.00 0.00 0.65 0.62 1.00 0.02 0.04 0.64 0.00 0.00 1.00
F2*  0.55 0.00 1.00 0.00 0.55 0.56 0.02 1.00 0.00 0.55 1.00 0.00 0.00
F3*  0.54 0.00 0.00 1.00 0.52 0.55 0.03 0.00 1.00 0.54 0.00 1.00 0.00
h2   1.00 0.65 0.55 0.52 1.00 1.00 0.68 0.57 0.55 1.00 0.55 0.52 0.64
g    1.00 0.62 0.56 0.55 1.00 1.00 0.65 0.58 0.57 1.00 0.56 0.55 0.62
F1*  0.67 1.00 0.02 0.03 0.68 0.65 1.00 0.04 0.07 0.67 0.02 0.03 1.00
F2*  0.56 0.02 1.00 0.00 0.57 0.58 0.04 1.00 0.00 0.56 1.00 0.00 0.02
F3*  0.57 0.04 0.00 1.00 0.55 0.57 0.07 0.00 1.00 0.57 0.00 1.00 0.03
     1.00 0.64 0.55 0.54 1.00 1.00 0.67 0.56 0.57 1.00 0.54 0.54 0.63
     0.54 0.00 1.00 0.00 0.55 0.56 0.02 1.00 0.00 0.54 1.00 0.00 0.00
     0.54 0.00 0.00 1.00 0.52 0.55 0.03 0.00 1.00 0.54 0.00 1.00 0.00
     0.63 1.00 0.00 0.00 0.64 0.62 1.00 0.02 0.03 0.63 0.00 0.00 1.00
```

6 Summary

In the modern era of computation, there is little justification for continuing with procedures that were developed as short-cuts 80 years ago (Revelle et al., 2020), To find ω_h, α, and ω_t is very easy using the open source statistical system (R) as well as the omega functions in the *psych* package.

7 System Info

When running any R package, it is useful to find out the session information to see if you have the most recent releases.

R code

```
> sessionInfo()
```

```
R version 4.1.1 (2021-08-10)
Platform: x86_64-apple-darwin17.0 (64-bit)
Running under: macOS Big Sur 10.16

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRlapack.dylib

locale:
[1] C

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] psychTools_2.1.12 psych_2.1.12
```

```
loaded via a namespace (and not attached):
 [1] compiler_4.1.1      parallel_4.1.1      tools_4.1.1      foreign_0.8-81      lavaan_0.6-7
 [8] mnormt_2.0.2        grid_4.1.1          pbivnorm_0.6.0   knitr_1.33          xfun_0.24
[15] lattice_0.20-44
```

References

Azzalini, A. and Genz, A. (2016). *The R package mnormt: The multivariate normal and t distributions (version 1.5-5).*

Bernaards, C. and Jennrich, R. (2005). Gradient projection algorithms and software for arbitrary rotation criteria in factor analysis. *Educational and Psychological Measurement*, 65(5):676–696.

Cooksey, R. and Soutar, G. (2006). Coefficient beta and hierarchical item clustering - an analytical procedure for establishing and displaying the dimensionality and homogeneity of summated scales. *Organizational Research Methods*, 9:78–98.

Cronbach, L. J. (1951). Coefficient alpha and the internal structure of tests. *Psychometrika*, 16:297–334.

Fox, J., Nie, Z., and Byrnes, J. (2013). *sem: Structural Equation Models*. R package version 3.1-3.

Guttman, L. (1945). A basis for analyzing test-retest reliability. *Psychometrika*, 10(4):255–282.

Jensen, A. R. and Weng, L.-J. (1994). What is a good g? *Intelligence*, 18(3):231–258.

Kuder, G. and Richardson, M. (1937). The theory of the estimation of test reliability. *Psychometrika*, 2(3):151–160.

McDonald, R. P. (1999). *Test theory: A unified treatment*. L. Erlbaum Associates, Mahwah, N.J.

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Revelle, W. (1979). Hierarchical cluster-analysis and the internal structure of tests. *Multivariate Behavioral Research*, 14(1):57–74.

Revelle, W. (2020). *psych: Procedures for Personality and Psychological Research*. Northwestern University, Evanston, https://CRAN.r-project.org/package=psych. R package version 2.0.8.

Revelle, W. and Condon, D. M. (2018). Reliability. In Irwing, P., Booth, T., and Hughes, D. J., editors, *The Wiley Handbook of Psychometric Testing: A Multidisciplinary Reference on Survey, Scale and Test Development*. John Wily & Sons, London.

Revelle, W. and Condon, D. M. (2019). Reliability from α to ω: A tutorial. *Psychological Assessment* 31 (12) p 1395-1411.

Revelle, W. and Elleman, L.G. and Hall, A. (2020). Statistical analyses and computer programming in personality. In Corr, P.J. editor, *The Cambridge University Press Handbook of Personality*. Cambridge University Press.

Revelle, W. and Zinbarg, R. E. (2009). Coefficients alpha, beta, omega and the glb: comments on Sijtsma. *Psychometrika*, 74(1):145–154.

Rosseel, Y. (2012). lavaan: An R package for structural equation modeling. *Journal of Statistical Software*, 48(2):1–36.

Waller, N. G. (2017). Direct Schmid-Leiman transformations and rank-deficient loadings matrices. *Psychometrika*.

Zinbarg, R. E., Revelle, W., Yovel, I., and Li, W. (2005). Cronbach's α, Revelle's β, and McDonald's ω_H: Their relations with each other and two alternative conceptualizations of reliability. *Psychometrika*, 70(1):123–133.

Zinbarg, R. E., Yovel, I., Revelle, W., and McDonald, R. P. (2006). Estimating generalizability to a latent variable common to all of a scale's indicators: A comparison of estimators for ω_h. *Applied Psychological Measurement*, 30(2):121–144.