Using R and the *psych* package to find ω

William Revelle
Department of Psychology
Northwestern University

February 25, 2013

Contents

1 Overview of this and related documents

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 in May, 2012 to the Association of Psychological Science `http://personality-project.org/r/aps/aps-short.pdf` as well as a short guide to R `http://personality-project.org/r/` for psychologists and the vignette for the *psych* package `http://cran.r-project.org/web/packages/psych/vignettes/overview.pdf`.

McDonald has proposed coefficient omega (hierarchical) (ω_h) as an estimate of the general factor saturation of a test. Zinbarg et al. (2005) `http://personality-project.org/revelle/publications/zinbarg.revelle.pmet.05.pdf` 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) `http://personality-project.org/revelle/publications/revelle.zinbarg.08.pdf`).

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, 2013).

Assuming that your data are in a file called `my.data`, and that you have the necessary packages installed, to find ω_h requires two lines of code:

- library(psych)
- omega(my.data)

The resulting output will be both graphical (see Figure 1) and textual.

This guide helps the naive R user to issue those two lines.

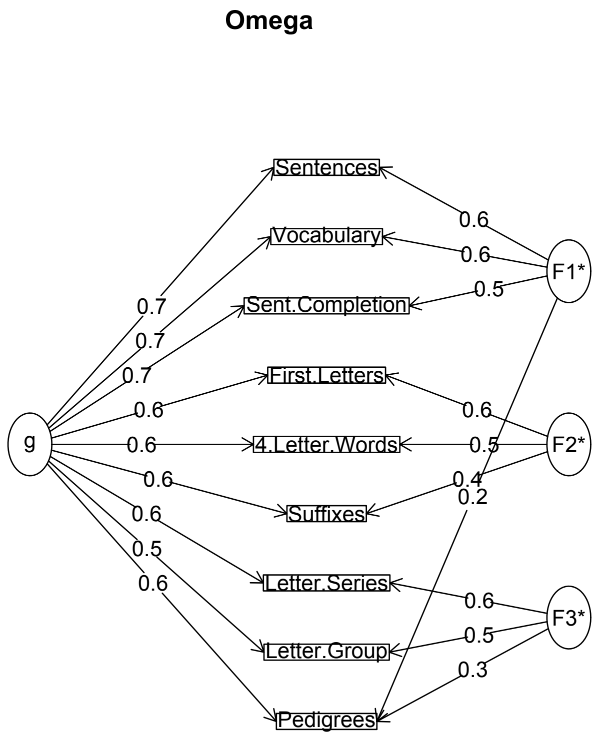

Figure 1: ω_h is a reliability estimate of the general factor of a set of variables. It is based upon the correlation of lower order factors. It may be found in R by using the `omega` function which is part of the *psych* package. The figure shows a solution for the `Thurstone` 9 variable data set.

2 Install R and relevant packages

To find ωh in R obviously requires installing R on your computer. This is very easy to do (see section 2.1.1) and needs to be done once.

The power of R is in the supplemental *packages*. At a minimum, you will need to install three package (*psych* (Revelle, 2013), *GPArotation* (Bernaards and Jennrich, 2005) and *MASS* (Venables and Ripley, 2002). With these three packages, you will be be able to find ω_h using Exploratory Factor Analysis. If you want to find it using Confirmatory Factor Analysis, you will also need to add the *sem* (Fox et al., 2013) and *matrixcalc* (Novomestky, 2012) packages. 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.

2.1 Install R for the first time

1. Download from R Cran (`http://cran.r-project.org/`) (see section 2.1.1)
   - Choose appropriate operating system and download compiled R
2. Install R (current version is 2.15.2)
3. Start R
4. Add useful packages (just need to do this once) (see section 2.1.2)
   (a) `install.packages(c("psych","GPArotation","MASS"))` #the minimum requirement
   (b) or if you want to do CFA

   `install.packages(c("psych","GPArotation","MASS","sem","matrixcalc"))`
5. or if you want to install the psychometric task views
   (a) `install.packages("ctv")` #this downloads the task view package
   (b) library(ctv) #this activates the ctv package
   (c) `install.views("Psychometrics")` #among others
   (d) Take a 5 minute break
6. Activate the package(s) you want to use (e.g., *psych*)
   - library(psych) #necessary for finding omega

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 1.3.1

7. Use R

2.1.1 Install R

First go to the Comprehensive R Archive Network (CRAN) at http://cran.r-project.org: (Figure 2)

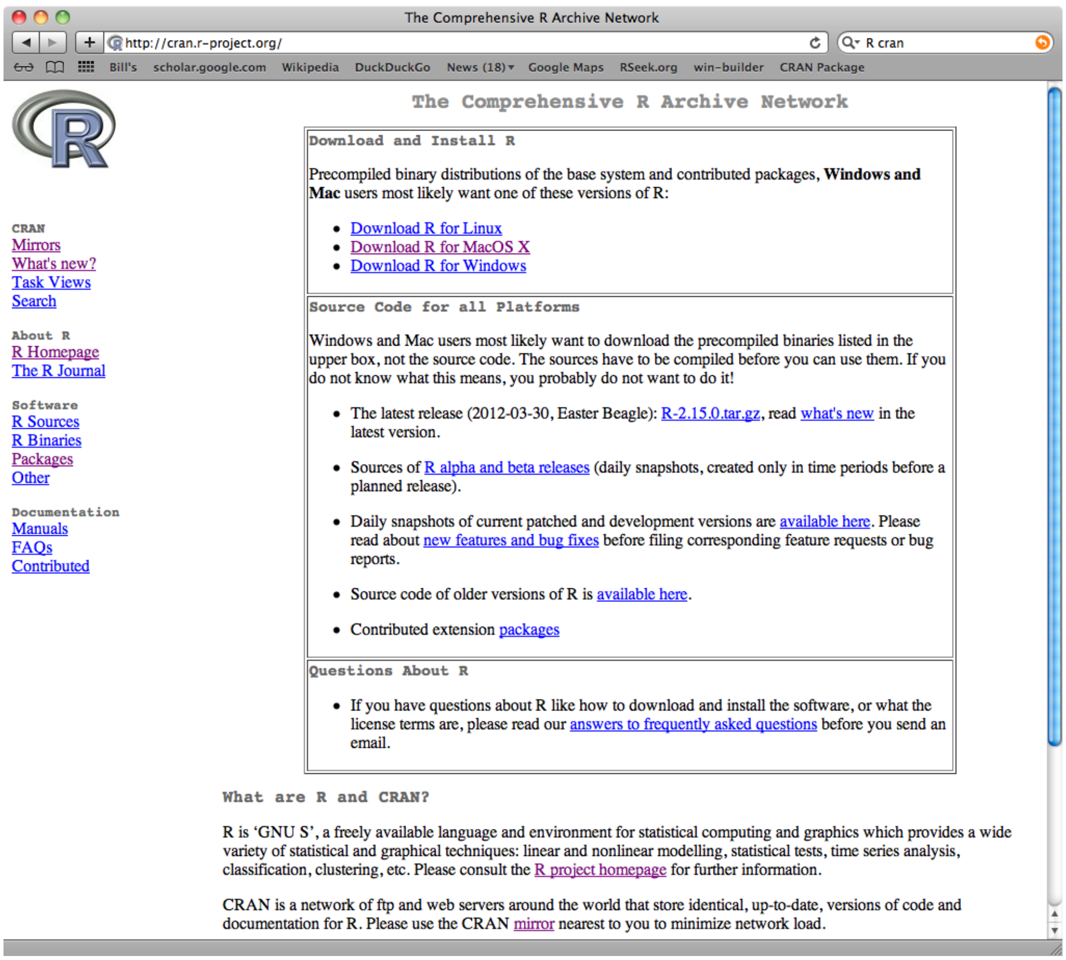

Figure 2: The basic CRAN window allows you choose your operating system. Comprehensive R Archive Network (CRAN) is found at http://cran.r-project.org:

Choose your operating system and then download and install the appropriate version

For a PC: (Figure 3)

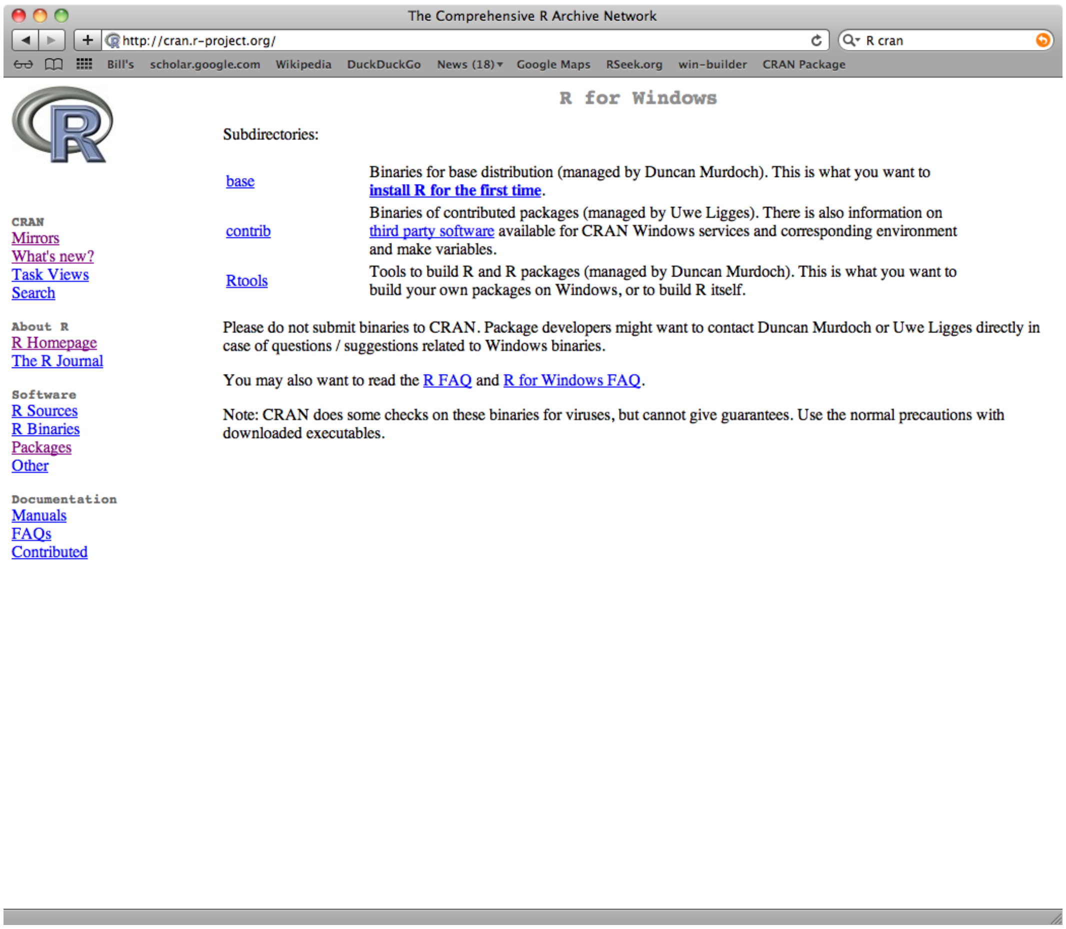

Figure 3: On a PC you want to choose the base system

Download and install the appropriate version – PC

For a Mac: download and install the appropriate version – Mac (Figure 5)

Starting R on a PC

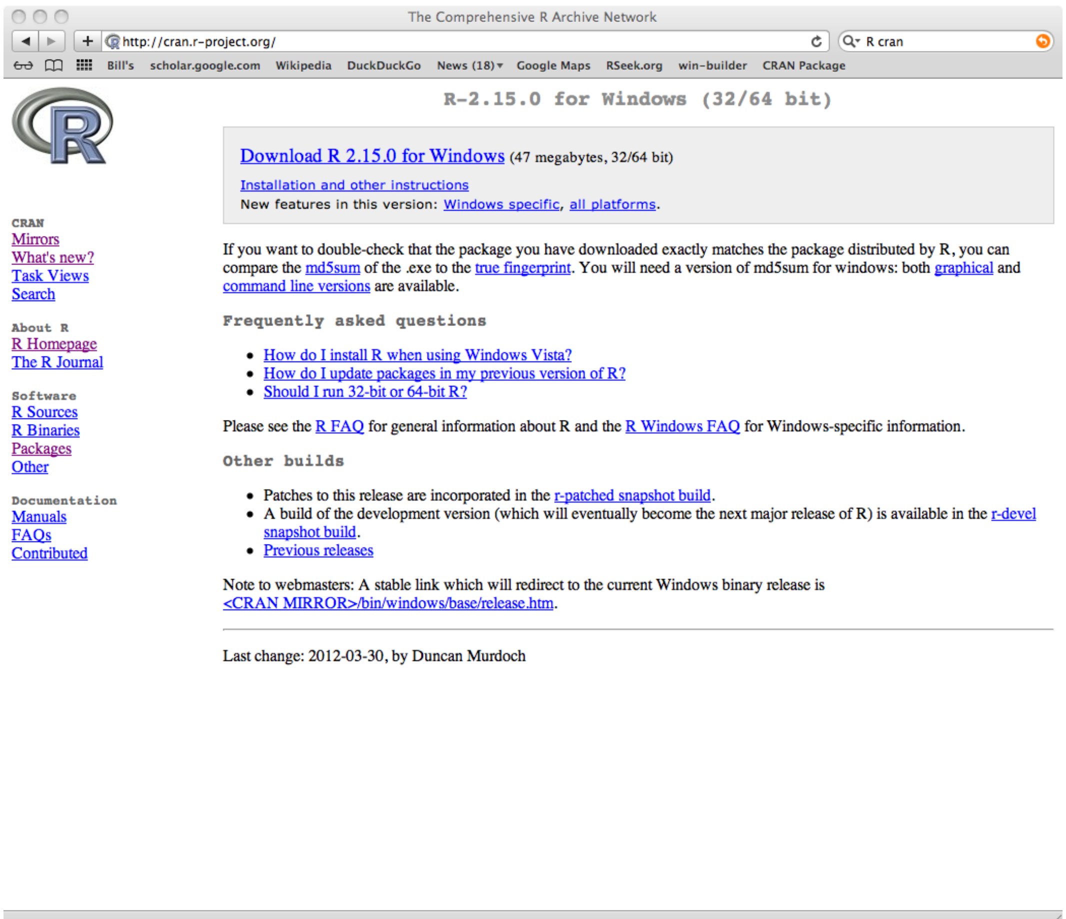

Figure 4: Download the Windows version

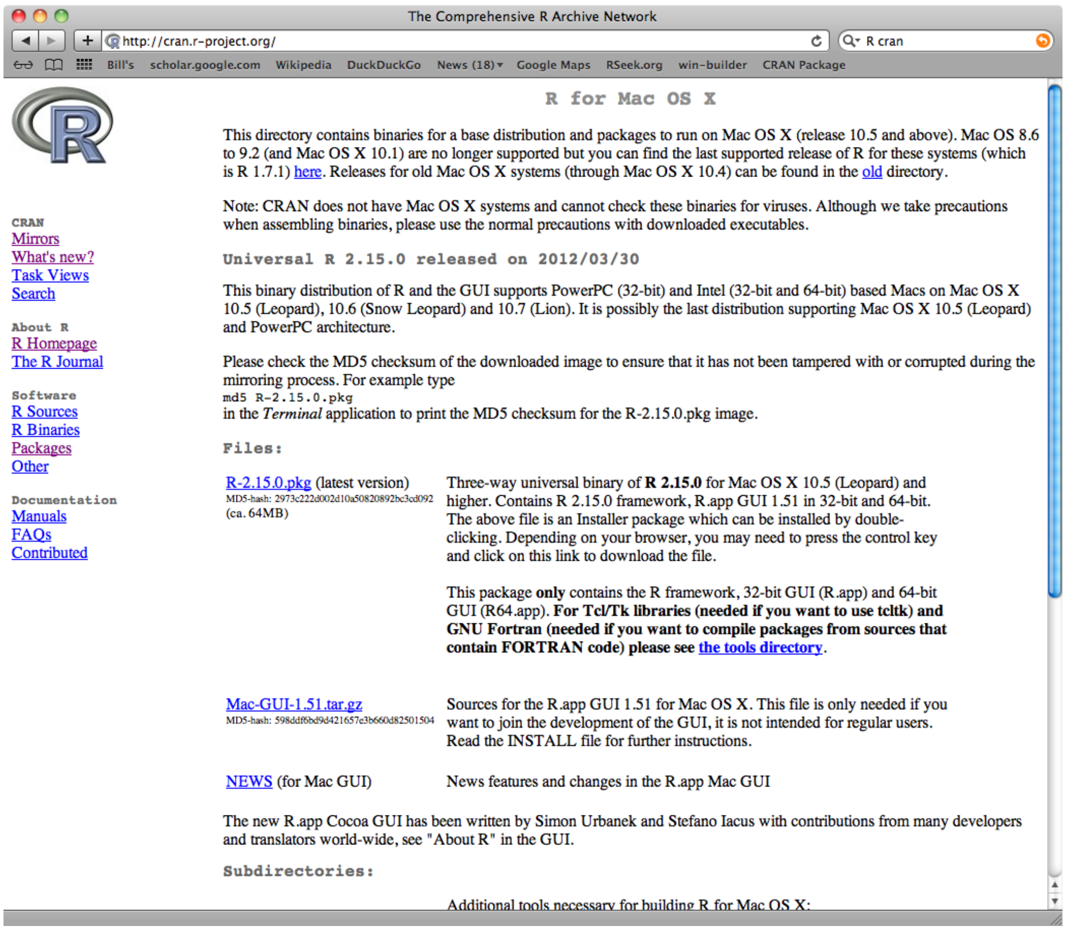

Figure 5: For the Mac, you want to choose the latest version which includes the GUI as well as the 32 and 64 bit versions.

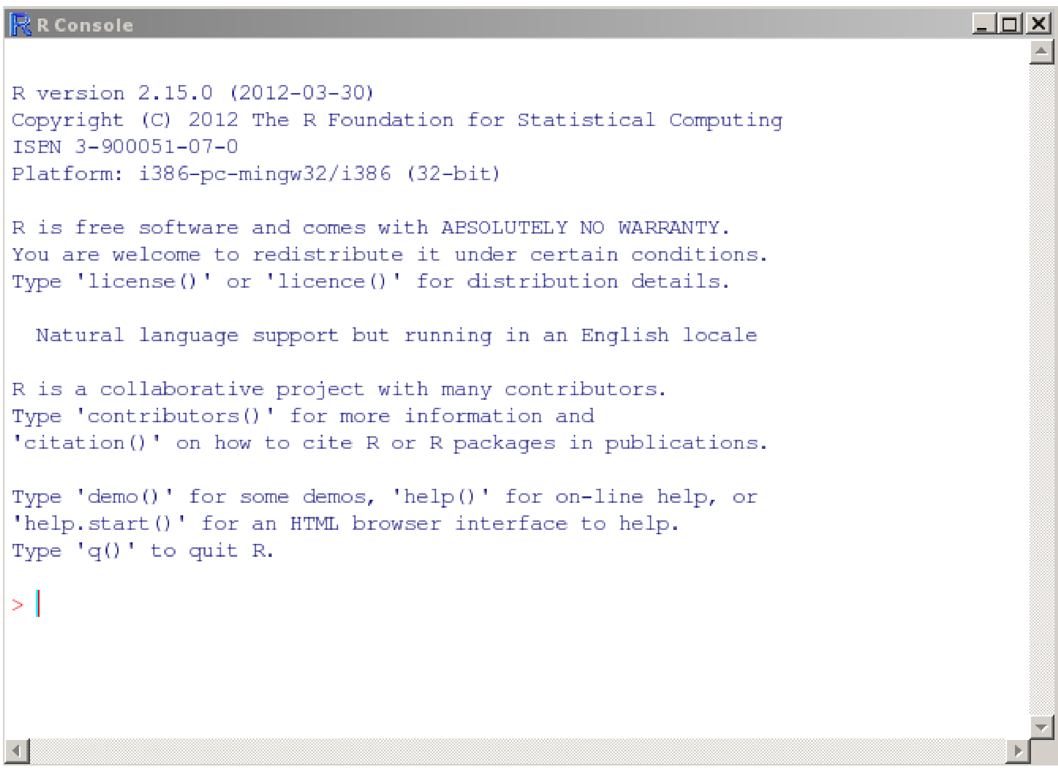

Figure 6: The startup screen on a PC

Start up R and get ready to play (Mac version)

```
R version 2.15.0 Patched (2012-03-30 r58887) -- "Easter Beagle"
Copyright (C) 2012 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: x86_64-apple-darwin9.8.0/x86_64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[R.app GUI 1.43 (5920) x86_64-apple-darwin9.8.0]

[Workspace restored from /Users/revelle/.RData]
[History restored from /Users/revelle/.Rapp.history]
>    # >  is the prompt for all commands    #is for comments
```

2.1.2 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, that application is for doing the kind of basic data analysis and psychometric analysis that personality psychologists find particularly useful.

You may either install the minimum set of packages necessary to do the analysis using an EFA approach (recommended) or a few more packages to do both the EFA and a CFA approach. It is also possible to add many psychometrically relevant packages all at once by using the "task views" approach.

Install the minimum set

- `install.packages(c("psych","GPArotation","MASS"))`

Install the *sem* package as well

- `install.packages(c("psych","GPArotation","MASS","sem","matrixcalc"))`

Install all the psychometric packages from the "psychometrics" task view.

- `install.packages("ctv")` #this downloads the task view package
- library(ctv) #this activates the ctv package
- `install.views("Psychometrics")` #among others
- Take a 5 minute break

You are almost ready. But first you need to make the *psych* package active. You only need to do this once per session.

- library(psych) #necessary for finding omega

You are almost there. But first you need to read in your data from your data file.

3 Read in the data

There are of course many ways to enter data into R. Reading from a local file using `read.table` is perhaps the most preferred. You first need to find the file and then read it. This can be done with the `file.choose` and `read.table` functions:

```
file.name <- file.choose()
my.data <- read.table(file.name)
```

`file.choose` opens a search window on your system just like any open file command does. It doesn't actually read the file, it just finds the file. The read command is also necessary.

3.1 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 `read.table` and specifying the input file as "clipboard" (PCs) or "pipe(pbpaste)" (Macs). Alternatively, the `read.clipboard` set of functions are perhaps more user friendly:

`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

```
> 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.

```
> 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).

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

3.2 Import from an SPSS or SAS file

To read data from an SPSS, SAS, or Systat file, you must use the *foreign* package. This might need to be installed (see 2.1.2 for instructions for installing packages).

`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)
```

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?

An example of reading from an SPSS file and then describing the data set to make it looks ok.

```
> library(foreign)
> datafilename <- "http://personality-project.org/r/datasets/finkel.sav"
>  eli <- read.spss(datafilename,to.data.frame=TRUE,
                                     use.value.labels=FALSE)

> 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
```

4 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.

```
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 kurtosis   se
gender      1 700   1.65   0.48      2    1.68   0.00   1   2     1 -0.61    -1.62 0.02
education   2 700   3.16   1.43      3    3.31   1.48   0   5     5 -0.68    -0.07 0.05
age         3 700  25.59   9.50     22   23.86   5.93  13  65    52  1.64     2.42 0.36
ACT         4 700  28.55   4.82     29   28.84   4.45   3  36    33 -0.66     0.53 0.18
SATV        5 700 612.23 112.90    620  619.45 118.61 200 800   600 -0.64     0.33 4.27
SATQ        6 687 610.22 115.64    620  617.25 118.61 200 800   600 -0.59    -0.02 4.41
```

There are, of course, all kinds of things you could do with your data at this point, but read about them in the vignette for the *psych* package http://cran.r-project.org/web/packages/psych/vignettes/overview.pdf.

5 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. 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`.

5.1 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) http://personality-project.org/revelle/publications/zinbarg.revelle.pmet.05.pdf 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) http://personality-project.org/revelle/publications/revelle.zinbarg.08.pdf).

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).

β, an alternative to ω, is defined as the worst split half reliability. 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).

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 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, $\vec{g}$, that due to a set of group factors, $\vec{f}$, (factors common to some but not all of the items), specific factors, $\vec{s}$ unique to each item, and $\vec{e}$, 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 $\vec{x} = \vec{c}g + \vec{A}f + \vec{D}s + \vec{e}$ 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

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

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.

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

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

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

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.

5.2 Using the `omega` function

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.

```
> omega(Thurstone)
```

```
Omega
Call: omega(m = Thurstone)
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  0.57              0.82 0.18 0.61
Vocabulary      0.73  0.55              0.84 0.16 0.63
Sent.Completion 0.68  0.52              0.73 0.27 0.63
First.Letters   0.65        0.56        0.73 0.27 0.57
4.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              0.61 0.72 0.28 0.48
Pedigrees       0.58  0.23        0.34 0.50 0.50 0.66
Letter.Group    0.54              0.46 0.53 0.47 0.56

With eigenvalues of:
   g  F1*  F2*  F3*
3.58 0.96 0.74 0.71

general/max  3.71   max/min =   1.35
mean percent general =  0.6    with sd =  0.05 and cv of  0.09

The degrees of freedom are 12  and the fit is  0.01

The root mean square of the residuals is  0
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  0.1
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 0.52 0.56
Minimum correlation of factor score estimates 0.49 0.08 0.03 0.11
```

5.3 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 *sem* package. These results can produce slightly different estimates of ω_h, primarily because cross loadings are modeled as part of the general factor.

```
> omegaSem(r9,n.obs=500)

Call: omegaSem(m = r9, n.obs = 500)
Omega
Call: omega(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)
Alpha:                 0.75
G.6:                   0.74
Omega Hierarchical:    0.66
Omega H asymptotic:    0.84
Omega Total            0.78

Schmid Leiman Factor loadings greater than  0.2
      g   F1*   F2*   F3*   h2   u2   p2
V1 0.70                   0.53 0.47 0.93
V2 0.70                   0.52 0.48 0.94
V3 0.54                   0.32 0.68 0.91
V4 0.53  0.46             0.50 0.50 0.57
V5 0.44  0.44             0.39 0.61 0.50
```

```
V6 0.40  0.32            0.26 0.74 0.59
V7 0.31             0.31 0.21 0.79 0.48
V8 0.34             0.44 0.30 0.70 0.37
V9 0.24             0.36 0.19 0.81 0.32

With eigenvalues of:
   g  F1*  F2*  F3*
2.18 0.52 0.08 0.44

general/max  4.21   max/min =   6.17
mean percent general =  0.62    with sd =  0.24 and cv of  0.39

The degrees of freedom are 12  and the fit is  0.03
The number of observations was  500  with Chi Square =  14.23  with prob <  0.29
The root mean square of the residuals is  0.01
The df corrected root mean square of the residuals is  0.03
RMSEA index =  0.02  and the 90 % confidence intervals are  NA 0.052
BIC =  -60.35

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  0.21
The number of observations was  500  with Chi Square =  103.64  with prob <  6.4e-11
The root mean square of the residuals is  0.05
The df corrected root mean square of the residuals is  0.08

RMSEA index =  0.076  and the 90 % confidence intervals are  0.06 0.091
BIC =  -64.15

Measures of factor score adequacy
                                                 g   F1*   F2*   F3*
Correlation of scores with factors            0.86  0.63  0.25  0.59
Multiple R square of scores with factors      0.74  0.39  0.06  0.35
Minimum correlation of factor score estimates 0.48 -0.21 -0.88 -0.30

 Omega Hierarchical from a confirmatory model using sem =  0.68
 Omega Total  from a confirmatory model using sem =  0.78
With loadings of
      g  F1*  F2*  F3*   h2   u2
V1 0.73                0.54 0.46
V2 0.68      0.29      0.54 0.46
V3 0.51      0.22      0.31 0.69
V4 0.54 0.47           0.51 0.49
V5 0.45 0.42           0.38 0.62
V6 0.39 0.31           0.25 0.75
V7 0.34           0.34 0.23 0.77
V8 0.36           0.39 0.28 0.72
V9 0.26           0.33 0.18 0.82

With eigenvalues of:
   g  F1*  F2*  F3*
2.21 0.49 0.14 0.38
```

References

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.

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.

Fox, J., Nie, Z., and Byrnes, J. (2013). *sem: Structural Equation Models*. R package version 3.1-0.

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

Novomestky, F. (2012). *matrixcalc: Collection of functions for matrix calculations*. R package version 1.0-3.

Revelle, W. (2013). *psych: Procedures for Personality and Psychological Research*. Northwestern University, Evanston, http://cran.r-project.org/web/packages/psych/. R package version 1.3.1.

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

Venables, W. N. and Ripley, B. D. (2002). *Modern Applied Statistics with S*. Springer, New York, fourth edition. ISBN 0-387-95457-0.

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.