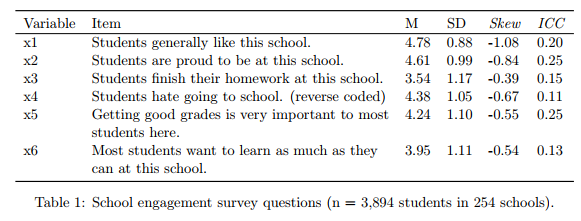
**DRAFT: Notes on Internal Consistency and Reliability [see McNeish 2017 & Streiner, 2003]**

**Francis Huang / 2017.10.03 / updated 2018.02.26**

::: install the package “userfriendlyscience”🡪 it installs several packages so do this while I am discussing.

* Some observed score or scale (X) for a trait or construct has two components: the true component (T) and the error component (E) such that X = T + E
* A score is more reliable when the T accounts for the greater proportion of the overall score (e.g., a score made up of error is silly and is pretty random)
* Reliability is the ratio of the true score to the observed score variance or Var(T)/Var(X)
* This can be interpreted as a correlation between two consecutive administrations (assuming we wipe the memory of the person responding).
* Historically, the way of getting reliability required multiple administrations (e.g., test-retest)
* Computing it with one administration of the test/scale is thus much appreciated! (Don’t need to administer a test twice!). Idea is if participants provide similar answers to a set of items, their responses should generalize to other items in a similar domain 🡪 thus items would have high internal consistency

Let’s take an example: see: <http://faculty.missouri.edu/huangf/data/mcfa/MCFAinRHUANG.pdf>



> library(psych)

> x <- read.csv("http://faculty.missouri.edu/huangf/data/mcfa/raw.csv")

> str(x) #last one is an id variable, let's take it out

'data.frame': 3894 obs. of 7 variables:

$ x1 : int 4 6 5 4 4 4 5 3 4 4 ...

$ x2 : int 4 6 5 4 4 4 4 2 4 4 ...

$ x3 : int 3 4 2 5 4 2 3 2 2 2 ...

$ x4 : int 4 6 6 5 4 4 5 3 4 5 ...

$ x5 : int 4 4 4 3 3 2 3 3 1 2 ...

$ x6 : int 3 4 4 3 2 2 3 2 1 2 ...

$ sid: int 55 55 55 55 55 55 41 41 41 41 ...

>

> dat <- x[, -7] #remove the schoolid (sid)

> head(dat)

x1 x2 x3 x4 x5 x6

1 4 4 3 4 4 3

2 6 6 4 6 4 4

3 5 5 2 6 4 4

4 4 4 5 5 3 3

5 4 4 4 4 3 2

6 4 4 2 4 2 2

> describe(dat)

vars n mean sd median trimmed mad min max range skew kurtosis se

x1 1 3894 4.78 0.88 5 4.85 0.00 1 6 5 -1.08 2.01 0.01

x2 2 3894 4.61 0.99 5 4.69 1.48 1 6 5 -0.84 1.00 0.02

x3 3 3894 3.54 1.17 4 3.60 1.48 1 6 5 -0.39 -0.53 0.02

x4 4 3894 4.38 1.05 5 4.42 1.48 1 6 5 -0.67 0.15 0.02

x5 5 3894 4.24 1.10 4 4.30 1.48 1 6 5 -0.55 0.11 0.02

x6 6 3894 3.95 1.11 4 4.03 1.48 1 6 5 -0.54 -0.04 0.02

> options(digits = 2)

> cor(dat)

x1 x2 x3 x4 x5 x6

x1 1.00 0.82 0.43 0.47 0.53 0.54

x2 0.82 1.00 0.43 0.44 0.55 0.56

x3 0.43 0.43 1.00 0.28 0.58 0.60

x4 0.47 0.44 0.28 1.00 0.37 0.37

x5 0.53 0.55 0.58 0.37 1.00 0.74

x6 0.54 0.56 0.60 0.37 0.74 1.00

### how many factors?

fa.parallel(dat, SMC = T)

# results support two factors

scree(dat) #seems to flatten out at three, so 2 factors

efa.res <- fa(dat, nfactors = 2, fm = 'pa')

efa.res

#two factors are correlated, .69

#factor 1 is x1, x2, x4

#factor 2 is x3, x5, x6

Factor Analysis using method = pa

Call: fa(r = dat, nfactors = 2, fm = "pa")

Standardized loadings (pattern matrix) based upon correlation matrix

PA1 PA2 h2 u2 com

x1 -0.05 0.97 0.87 0.13 1.0

x2 0.07 0.83 0.77 0.23 1.0

x3 0.69 -0.01 0.47 0.53 1.0

x4 0.14 0.41 0.27 0.73 1.2

x5 0.85 0.01 0.72 0.28 1.0

x6 0.86 0.00 0.75 0.25 1.0

PA1 PA2

SS loadings 2.02 1.83

Proportion Var 0.34 0.31

Cumulative Var 0.34 0.64

Proportion Explained 0.52 0.48

Cumulative Proportion 0.52 1.00

With factor correlations of

PA1 PA2

PA1 1.00 0.69

PA2 0.69 1.00

So how do we now get the reliabilities?

> names(dat)

[1] "x1" "x2" "x3" "x4" "x5" "x6"

> options(digits = 3)

> cov(dat[,c(1,2,4)]) #selecting vars 1,2,4

x1 x2 x4

x1 0.769 0.706 0.436

x2 0.706 0.975 0.457

x4 0.436 0.457 1.102

> sum(cov(dat[,c(1,2,4)]) )

[1] 6.04 #this way

> f1 <- dat$x1 + dat$x2 + dat$x4

> var(f1)

[1] 6.04 #or that way

So how do we now get the reliabilities? NOTE: the reliabilities are estimated using the correlation matrix—which shows how related our variables are:

Cronbach’s alpha = where is the mean covariate/correlation of items on the scale. is the total variance of the scale.

<http://faculty.missouri.edu/huangf/data/mcfa/MCFAinRHUANG.pdf> (p. 16)

* The numerator is the product of the square of the number of items in the scale and the average of the unique covariance elements.
* The denominator is merely the sum of all the elements within the covariance matrix or summing together all the variances and two times the covariance elements.

> # so alpha is then [see where all the pieces come from]

> (.706 + .436 + .457)/3 #average covariance

[1] 0.533

> 3^2 \* .533

[1] 4.8

> 4.8/6.04 #this is alpha🡪 see where we got this

[1] 0.795

> alpha(dat[,c(1,2,4)])

Reliability analysis

Call: alpha(x = dat[, c(1, 2, 4)])

raw\_alpha std.alpha G6(smc) average\_r S/N ase mean sd

0.79 0.8 0.78 0.58 4.1 0.0061 4.6 0.82

Reliability if an item is dropped: #do not just use this to determine which item fits better

raw\_alpha std.alpha G6(smc) average\_r S/N alpha se

x1 0.61 0.61 0.44 0.44 1.6 0.0124

x2 0.64 0.64 0.47 0.47 1.8 0.0114

x4 0.90 0.90 0.82 0.82 8.9 0.0033

Can also do this simply using the correlation matrix.

Image 

From: https://stats.idre.ucla.edu/spss/faq/what-does-cronbachs-alpha-mean/

N = number of items, c = average correlation/covariance, v = average variance (if using a correlation matrix, this is just 1, right?)

> x <- read.csv("http://faculty.missouri.edu/huangf/data/mcfa/raw.csv")

> names(x)

[1] "x1" "x2" "x3" "x4" "x5" "x6" "sid"

> x <- x[,-7]

>

>

> f1 <- x[,c(1,2,4)]

> cov(f1)

x1 x2 x4

x1 0.7685493 0.7060844 0.4360729

x2 0.7060844 0.9746548 0.4566123

x4 0.4360729 0.4566123 1.1016003

> cor(f1)

x1 x2 x4

x1 1.0000000 0.8158219 0.4739267

x2 0.8158219 1.0000000 0.4406666

x4 0.4739267 0.4406666 1.0000000

>

> mc <-(.82 + .47 + .44)/3

> (3 \* mc ) / (1 + (2 \* mc))

[1] 0.8034056

>

> library(psych)

> alpha(f1)

Reliability analysis

Call: alpha(x = f1)

raw\_alpha std.alpha G6(smc) average\_r S/N ase mean sd

0.79 0.8 0.78 0.58 4.1 0.0061 4.6 0.82

lower alpha upper 95% confidence boundaries

0.78 0.79 0.81

Reliability if an item is dropped:

raw\_alpha std.alpha G6(smc) average\_r S/N alpha se

x1 0.61 0.61 0.44 0.44 1.6 0.0124

x2 0.64 0.64 0.47 0.47 1.8 0.0114

x4 0.90 0.90 0.82 0.82 8.9 0.0033

Item statistics

n raw.r std.r r.cor r.drop mean sd

x1 3894 0.89 0.90 0.88 0.75 4.8 0.88

x2 3894 0.88 0.89 0.86 0.71 4.6 0.99

x4 3894 0.77 0.75 0.51 0.48 4.4 1.05

Non missing response frequency for each item

1 2 3 4 5 6 miss

x1 0.00 0.02 0.04 0.21 0.55 0.16 0

x2 0.01 0.03 0.07 0.28 0.45 0.16 0

x4 0.01 0.04 0.15 0.26 0.44 0.10 0

>

> ####

>

> f2 <- x[,c(3,5,6)]

> alpha(f2)

Reliability analysis

Call: alpha(x = f2)

raw\_alpha std.alpha G6(smc) average\_r S/N ase mean sd

0.84 0.84 0.79 0.64 5.3 0.0045 3.9 0.98

lower alpha upper 95% confidence boundaries

0.83 0.84 0.85

Reliability if an item is dropped:

raw\_alpha std.alpha G6(smc) average\_r S/N alpha se

x3 0.85 0.85 0.74 0.74 5.6 0.0049

x5 0.75 0.75 0.60 0.60 3.0 0.0081

x6 0.74 0.74 0.58 0.58 2.8 0.0084

Item statistics

n raw.r std.r r.cor r.drop mean sd

x3 3894 0.84 0.83 0.68 0.63 3.5 1.2

x5 3894 0.88 0.89 0.82 0.74 4.2 1.1

x6 3894 0.89 0.89 0.83 0.75 3.9 1.1

Non missing response frequency for each item

1 2 3 4 5 6 miss

x3 0.05 0.16 0.21 0.37 0.19 0.02 0

x5 0.01 0.07 0.13 0.35 0.33 0.11 0

x6 0.03 0.09 0.17 0.38 0.29 0.05 0

>

> cor(f2)

x3 x5 x6

x3 1.0000000 0.5830168 0.5960668

x5 0.5830168 1.0000000 0.7362941

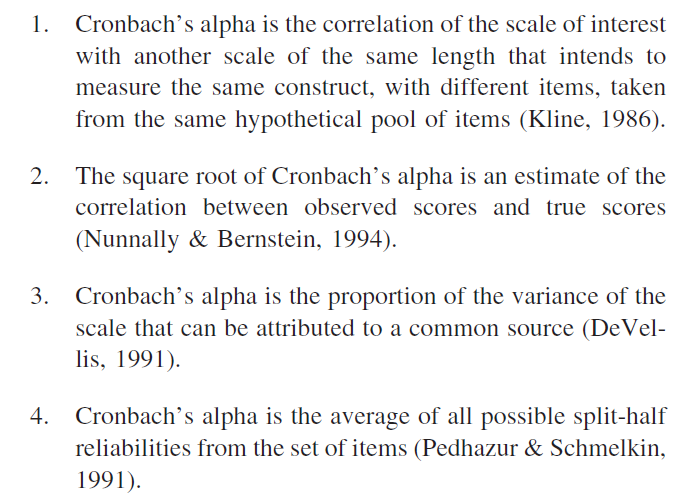
x6 0.5960668 0.7362941 1.0000000

> mc <- (.58 + .60 + .74) / 3

> (3 \* mc) / (1 + (2 \* mc))

[1] 0.8421053

What does alpha really mean? We say that it is a measure of how ‘items hang together’ (not very precise a definition, is it?): see McNeish article:



NOTE: So many articles have criticized alpha (just search for ‘problems with cronbach’s alpha’

What are the assumptions of Cronbach’s alpha:

1. Tau equivalence (every item contributes/loads in the same way)
2. Scale items are continuous and normally distributed
3. The errors of the items do not covary
4. The scale is unidimensional

1: Tau equivalence: look at our highlighted items

x1 -0.05 0.97 0.87 0.13 1.0

x2 0.07 0.83 0.77 0.23 1.0

x4 0.14 0.41 0.27 0.73 1.2

DO not all load with the same strength (x4 is only .41 for example):

Better yet, most scales are congeneric: means items load on the same construct but with different degrees of precision.

2. Continuous items: [NOTE: This is not exactly CORRECT—alpha can be used with dichotomous responses too! Formula KR-21 does just that which is just alpha]

NOTE: our items often are Likert scales (e.g., strongly disagree, disagree, agree, strongly agree). As such, we should actually use a polychoric correlation matrix for the factor analysis of these items. The polychoric correlation matrix can be automatically computed by R. It is not easy to do by hand. If we have categorical data can do:

fa.parallel.poly(dat, SMC = T) or

fa.poly(dat, nfactors = 2)

### THIS IS IMPORTANT

3. Errors do not covary: we can check this in a CFA, not with an EFA.

4. Unidimensional: measures one construct: this is verified through FA—this is not verified running alpha!!! This is a common mistake. Unidimensionality: degree which the items all measure the same underlying construct.

Internal consistency is NOT necessary for unidimensionality (it is not sufficient). Scales may be multidimensional (if everything loaded on one factor for example) and still have good internal consistency.

So what to use? Various alternatives, I tend to stick to Omega \*(see article from alpha to omega; Dunn, Baguley, & Brunsden, 2014) since it is similar to alpha but does not assume Tau equivalence (assumes that they are congeneric). Do not use the omega function in the psych package—it is computing something else and is higher. I tend to be more conservative…

A package called “userfriendlyscience” can compute all these for you: install it. (there’s a lot that it installs): It’s a package that installs several other packages. It takes a while but you only have to do it once. [that’s why I asked you to do it at the start of class!] I also use the **MBESS** package (function is ci.reliability)

> library(userfriendlyscience)

> factor1 <- dat[,c(1,2,4)]

> scaleStructure(dat = factor1)

Information about this analysis:

Dataframe: factor1

Items: all

Observations: 3894

Positive correlations: 3 out of 3 (100%)

Estimates assuming interval level:

Omega (total): 0.81

Omega (hierarchical): 0.02

Revelle's omega (total): 0.83

Greatest Lower Bound (GLB): 0.86

Coefficient H: 0.91

Cronbach's alpha: 0.79

Confidence intervals:

Omega (total): [0.8, 0.82]

Cronbach's alpha: [0.78, 0.8]

Estimates assuming ordinal level:

Ordinal Omega (total): 0.87

Ordinal Omega (hierarch.): 0.87

Ordinal Cronbach's alpha: 0.85

Confidence intervals:

Ordinal Omega (total): [0.86, 0.88]

Ordinal Cronbach's alpha: [0.84, 0.85]

NOTE: if you’re data are ordinal, the reliabilities are even better! NOTE: alpha is the lowest among those highlighted. So, why not use a method that is more applicable AND gives you better results?

NOTE: if userfriendlyscience is giving you errors, you can use the MBESS package (install it), and use the ci.reliability function.

|  |
| --- |
| > library(MBESS)  > factor1 <- dat[,c(1,2,4)]  > ci.reliability(factor1, type = 'omega')  $est  [1] 0.809819  $se  [1] 0.007174357  $ci.lower  [1] 0.7957575  $ci.upper  [1] 0.8238805  $conf.level  [1] 0.95  $type  [1] "omega"  $interval.type  [1] "robust maximum likelihood (wald ci)" |
|  |

**See paper of** (Dunn et al., 2014) **🡪 provides a tutorial on using MBESS (more than you might need). Also gives reasons why Omega is preferred:**

One measure that adheres to the congeneric model is that of *omega* (McDonald, 1999). Omega has been shown by

many researchers to be a more sensible index of internal consistency – both in relation to alpha and also when

compared to other alternatives (Zinbarg et al, 2005, 2006, 2007; Graham, 2006; Revelle & Zinbarg, 2005; Raykov,

1997). Zinbarg et al. (2005) report that even when the assumptions of the essentially tau-equivalent model are met,

omega performs at least as well as alpha. However, under violations of tau-equivalence – conditions likely to be the norm in psychology – omega outperforms alpha and is clearly the preferred choice

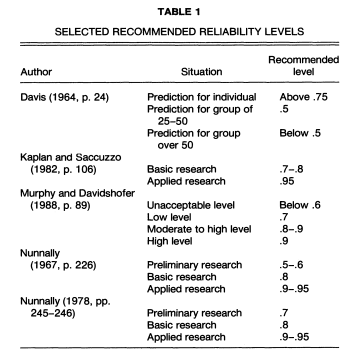
**So what is acceptable? [or what do you need if you want to publish with your measure]**

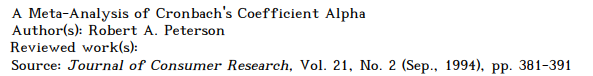
Generally (rule of thumb) acceptable reliabilities > .70 (often cited; Nunnaly, 1978, p. 245):

“…*one saves time and energy by working with instruments that have only modest reliability, for which purpose reliabilities of .70 or higher will suffice…”*

Others have said that .60 could be ok. Others will say .80. These are not based on any empirical guidelines but more conceptual understanding (i.e., how much of your measure is really error??)

Different guidelines have come out in the past (e.g., .6 is ok, .7 is ok) by the same authors (Nunnaly). When asked what he recommend, I think he answered (tongue in cheek): Depends what your coefficient is, if it’s low, cite the 1967 article. If it’s higher, cite the 1978 one [I AM JOKING! I am not recommending you do that]. IN THE END, use your judgement, but conventionally, the > .70 or .80 is often used (and you will not get questioned).





NOTE: if it’s too high (>.90; Streiner, 2003), you risk redundancy (you are asking almost the same thing!)

(Cronbach, 1951; McNeish, 2017; Peterson, 1994; Streiner, 2003)

**References**

Cronbach, L. J. (1951). Coefficient alpha and the internal structure of tests. *Psychometrika*, *16*, 297–334. https://doi.org/10.1007/BF02310555

Dunn, T. J., Baguley, T., & Brunsden, V. (2014). From alpha to omega: A practical solution to the pervasive problem of internal consistency estimation. *British Journal of Psychology*, *105*(3), 399–412.

McNeish, D. (2017). Thanks coefficient alpha, we’ll take it from here. Retrieved from http://psycnet.apa.org/psycinfo/2017-23572-001/

Peterson, R. A. (1994). A meta-analysis of Cronbach’s coefficient alpha. *Journal of Consumer Research*, *21*, 381–391.

Streiner, D. L. (2003). Being inconsistent about consistency: When coefficient alpha does and doesn’t matter. *Journal of Personality Assessment*, *80*, 217–222.