Psychology 350: Advanced statistics and programming in R
Reliability and Scale Construction

William Revelle

Department of Psychology
Northwestern University
Evanston, Illinois USA

April, 202r

Outline

Measures are befuddled with error
- Theory

Examples
- Two data sets
- Scoring the two sets

Reliability. part II
- `omega` takes into account the factor structure

Item Response Theory

Reliability

1. Reliability is a fundamental problem for measurement in all of science for “(a)ll measurement is befuddled by error" (p 294 McNemar, 1946).
2. Reliability is critical to the activity of measurement across many disciplines.
3. Reliability theory is not just for the psychometrician estimating latent variables, but also for the baseball manager trying to predict how well a high performing player will perform the next year, for accurately estimating agreement among doctors in patient diagnoses, and in evaluations of the extent to which stock market advisors under-perform the market.
4. Will we get the same result if we measure it again (or with an equivalent measure)?
5. What is the correlation of a scale with another scale said to measure the same construct?

Reliability: Correlation of a test with a test just like it

1. The fundamental question in reliability is to what extent do scores measured at one time and place with one instrument predict scores at another time and/or place and perhaps measured with a different instrument?
2. That is, given a person's score on test 1 at time 1, what score should be expected at a second measurement occasion (Revelle and Condon, 2019)?

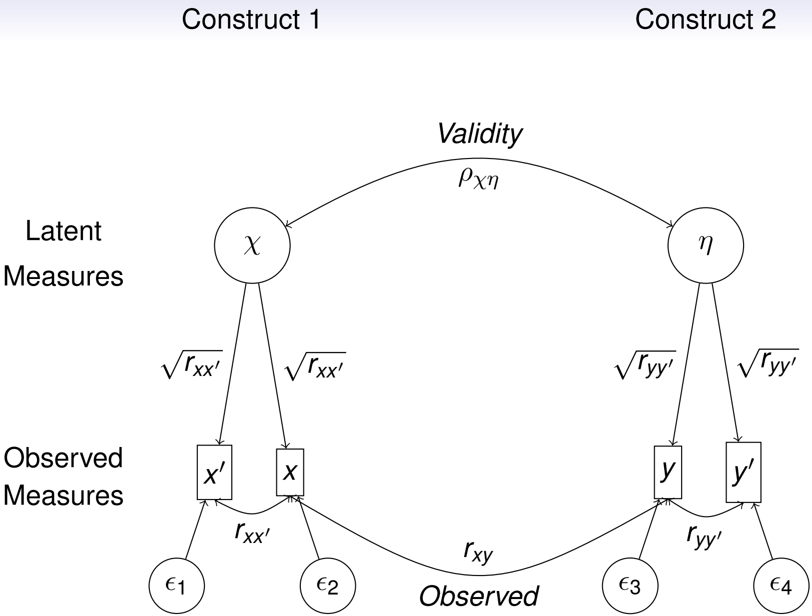
Reliability

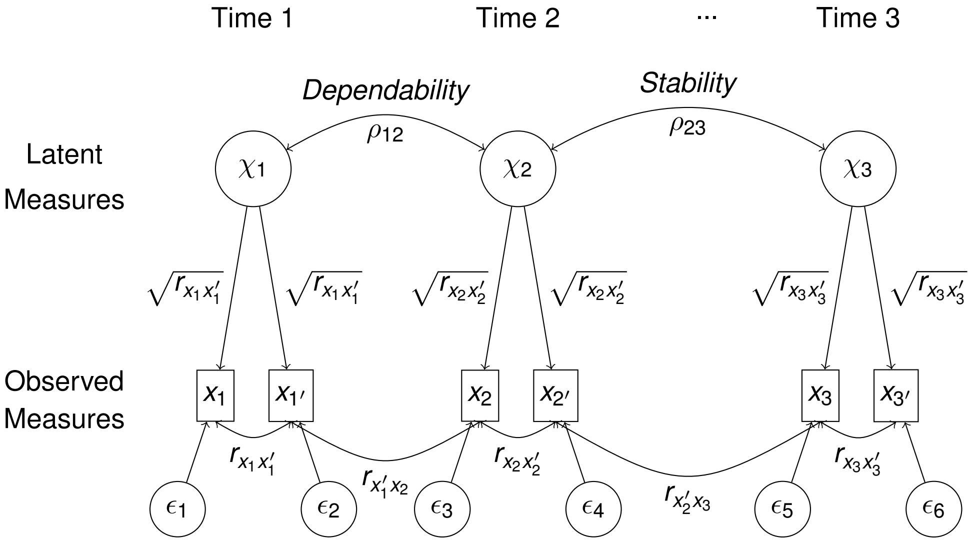
1. The basic concept of reliability seems to be very simple: observed scores reflect an unknown mixture of signal and noise (Spearman, 1904).
2. To detect the signal, we need to reduce the noise.
3. Reliability thus defined is a function of the ratio of signal to noise.
4. If an item is composed of signal and noise: $X = \chi + \epsilon$.
5. Then reliability is defined as the fraction of an observed score variance that was not error:

$$r_{xx} = \frac{V_X - \sigma^2_\epsilon}{V_X} = 1 - \frac{\sigma^2\epsilon}{V_X}. \tag{1}$$

6. And validity is the correlation between X and Y adjusted for the reliability of X and Y

$$\rho_{\chi\eta} = \frac{r_{xy}}{\sqrt{r_{xx}r_{yy}}}. \tag{2}$$


Construct 1
Construct 2
Validity
$\rho_{\chi\eta}$
Latent
Measures
χ
η
$\sqrt{r_{xx'}}$
$\sqrt{r_{xx'}}$
$\sqrt{r_{yy'}}$
$\sqrt{r_{yy'}}$
Observed
Measures
x'
x
y
y'
$r_{xx'}$
r_{xy}
$r_{yy'}$
ϵ_1
ϵ_2
Observed
ϵ_3
ϵ_4


Time 1
Time 2
...
Time 3
Dependability
Stability
ρ_{12}
ρ_{23}
Latent Measures
χ_1
χ_2
χ_3
$\sqrt{r_{x_1 x_1'}}$
$\sqrt{r_{x_2 x_2'}}$
$\sqrt{r_{x_3 x_3'}}$
Observed Measures
x_1
$x_{1'}$
x_2
$x_{2'}$
x_3
$x_{3'}$
$r_{x_1 x_1'}$
$r_{x_1' x_2}$
$r_{x_2 x_2'}$
$r_{x_2' x_3}$
$r_{x_3 x_3'}$
ϵ_1
ϵ_2
ϵ_3
ϵ_4
ϵ_5
ϵ_6

Scales as composites of items

1. The fundamental unit is the item
   - Perhaps an ability item (correct/incorrect)
   - An attitude item (agree/disagree or strongly agree somewhat agree, somewhat disagree, strongly disagree)
   - A preference item (Very much prefer, somwhat prefer, somwhat dislike, stongly dislike)
   - Psychological state (Currently feel ...)
   - Psychological trait (Often or usually feel ...)
2. But items are noisy (have a great deal of error)
3. Items may be thought of has having a common part (shared with other items) and specific part (not shared with other items) and error part (random mistakes)
4. Aggregate items to form scales to average out the error and the specific part.
5. What is the correlation of a scale with another scale said to measure the same construct?

The problem is how to find reliability

1. Two forms of same test
   - At same time (parallel forms)
   - At different times (test-retest)
     - After a short delay (dependabilty)
     - After a long delay (stability)
2. One test, multiple items
   - Forming parallel tests by split half correlation.
     - Random split
     - First half-second half
     - Odd/Even
     - All possible splits (to show variation) $C^n_{n/2} = \frac{n!}{(n/2)!(n-n/2)!} = \frac{n!}{((n/2)!(n/2)!}$
   - Based upon correlations of items (α, λ_3)
   - Based upon structure of items (ω_h, ω_t)

the `sai` and `tai` data sets in *psychTools*

1. State Anxiety was measured two-three times in 11 studies at the Personality-Motivation-Cognition laboratory. Here are item responses for 11 studies (9 repeated twice, 2 repeated three times).
2. In all studies, the first occasion was before a manipulation.
3. In some studies, caffeine, or movies or incentives were then given to some of the participants before the second and third STAI was given.
4. In addition, Trait measures are available and included in the tai data set (3032 subjects).

the `msqR` data set in *psychTools*

1. Emotions may be described either as discrete emotions or in dimensional terms.
2. The Motivational State Questionnaire (MSQ) was developed to study emotions in laboratory and field settings. The data can be well described in terms of a two dimensional solution of energy vs tiredness and tension versus calmness.
3. Alternatively, this space can be organized by the two dimensions of Positive Affect and Negative Affect.
4. Additional items include what time of day the data were collected and a few personality questionnaire scores.
5. 3032 unique participants took the MSQ at least once, 2753 at least twice, 446 three times, and 181 four times.
6. The 3032 participants also took the sai state anxiety inventory at the same time. Some studies manipulated arousal by caffeine, others manipulations included affect inducing movies.

Scoring multiple scales from a single set of items

1. Frequently we want give a set of items (a questionnaire) that includes items thought to measure different constructs.
2. We can specify those items for a particular scale by item #, or more readily, by item name.
3. We can form a list of such items for each scale.
4. Signify reverse scoring by a "-" sign.
5. Consider the keys for the the `sai` data.
6. First, show the column names

```
 colnames(sai)
 [1] "study"       "time"       "id"          "calm"
 [5] "secure"      "tense"      "regretful"   "at.ease"
 [9] "upset"       "worrying"   "rested"      "anxious"
[13] "comfortable" "confident"  "nervous"     "jittery"
[17] "high.strung" "relaxed"    "content"     "worried"
[21] "rattled"     "joyful"     "pleasant"
```

Using the sai data

Make a list of 3 scoring keys

R code

```
sai.keys <- list(sai = c("tense","regretful" , "upset", "worrying",
"anxious", "nervous" ,"jittery" ,"high.strung", "worried" ,"rattled",
"-calm", "-secure","-at.ease","-rested","-comfortable",
 "-confident" ,"-relaxed" , "-content" , "-joyful", "-pleasant"  ) ,
sai.p = c("calm","at.ease","rested","comfortable", "confident",
  "secure" ,"relaxed" ,   "content" , "joyful", "pleasant" ),
sai.n = c( "tense" , "anxious", "nervous" , "jittery" , "rattled",
         "upset", "worrying","worried","regretful" )
)
```

Show the list

```
sai.keys
$sai
 [1] "tense"         "regretful"     "upset"         "worrying"      "anxious"
 [6] "nervous"       "jittery"       "high.strung"   "worried"       "rattled"
[11] "-calm"         "-secure"       "-at.ease"      "-rested"       "-comfortable"
[16] "-confident"    "-relaxed"      "-content"      "-joyful"       "-pleasant"

$sai.p
 [1] "calm"          "at.ease"       "rested"        "comfortable" "confident"
 [6] "secure"        "relaxed"       "content"       "joyful"        "pleasant"

$sai.n
 [1] "tense"         "anxious"       "nervous"       "jittery"       "rattled"
 [6] "high.strung" "upset"         "worrying"      "worried"       "regretful"
```

Some items appeared in the sai and the msqR

R code

```
#these overlap with the msq
msq.items <- c("anxious", "at.ease" ,"calm", "confident","content",
 "jittery",  "nervous" ,  "relaxed" ,  "tense"  ,  "upset" )

sai.msq.keys <- list(
 pos =c( "at.ease" ,  "calm" , "confident",  "content","relaxed"),
  neg = c("anxious", "jittery", "nervous" ,"tense"  ,   "upset"),
anx = c("anxious", "jittery", "nervous" ,"tense", "upset",
     "-at.ease" ,  "-calm" ,  "-confident", "-content","-relaxed"))
```

```
 msq.items
 [1] "anxious"   "at.ease"   "calm"      "confident" "content"   "jittery"
 [7] "nervous"   "relaxed"   "tense"     "upset"
 sai.msq.keys
$pos
[1] "at.ease"   "calm"      "confident" "content"   "relaxed"

$neg
[1] "anxious" "jittery" "nervous" "tense"   "upset"

$anx
 [1] "anxious"    "jittery"    "nervous"    "tense"      "upset"      "-at.ease"
 [7] "-calm"      "-confident" "-content"   "-relaxed"
```

Select those subjects with the first time scores in `sai`

R code

```
dim(sai)
table(sai["time"])

 sai.once <- sai[sai[,"time"] ==1,]  #this chooses the subset
dim(sai.once)
```

```
> dim(sai)
[1] 5378   23
> table(sai["time"])
time
   1    2    3    4
3032 1229 1047   70
> sai.once <- sai[sai[,"time"] ==1,]
> dim(sai.once)
[1] 3032   23
```

Use the %in% command to find overlapping variables

R code

```
colnames(sai)
colnames(sai)%in% colnames(msq)  #returns a logical vector
colnames(sai)[ colnames(sai)%in% colnames(msq)] #use it
```

```
 colnames(sai)
 [1] "study"       "time"        "id"          "calm"
 [5] "secure"      "tense"       "regretful"   "at.ease"
 [9] "upset"       "worrying"    "rested"      "anxious"
[13] "comfortable" "confident"   "nervous"     "jittery"
[17] "high.strung" "relaxed"     "content"     "worried"
[21] "rattled"     "joyful"      "pleasant"

 colnames(sai)%in% colnames(msq)  #returns a logical vector
 [1] FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE  TRUE
[10] FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE
[19]  TRUE FALSE FALSE FALSE FALSE


colnames(sai)[ colnames(sai)%in% colnames(msq)] #use it
 [1] "calm"      "tense"     "at.ease"   "upset"
 [5] "anxious"   "confident" "nervous"   "jittery"
 [9] "relaxed"   "content"
```

Show the correlations of the sai items that overlap with the msq

R code

```
lowerCor(sai.once[msq.items])
#rearrange to make more sense of the  item
msq.items <- c( "at.ease" ,"calm", "confident","content",   "relaxed"
"anxious", "jittery",  "nervous" ,  "tense"  ,  "upset" )
lowerCor(sai.once[msq.items])
```

```
lowerCor(sai.once[msq.items])
          anxis at.es calm  cnfdn cntnt jttry nervs relxd tense upset
anxious    1.00
at.ease   -0.32  1.00
calm      -0.37  0.69  1.00
confident -0.05  0.45  0.34  1.00
content   -0.16  0.58  0.48  0.54  1.00
jittery    0.56 -0.34 -0.41 -0.06 -0.14  1.00
nervous    0.58 -0.38 -0.40 -0.15 -0.24  0.58  1.00
relaxed   -0.34  0.69  0.68  0.40  0.57 -0.40 -0.38  1.00
tense      0.57 -0.47 -0.49 -0.17 -0.31  0.54  0.63 -0.47  1.00
upset      0.32 -0.35 -0.30 -0.25 -0.35  0.22  0.39 -0.32  0.45  1.00

lowerCor(sai.once[msq.items])
          at.es calm  cnfdn cntnt relxd anxis jttry nervs tense upset
at.ease    1.00
calm       0.69  1.00
confident  0.45  0.34  1.00
content    0.58  0.48  0.54  1.00
relaxed    0.69  0.68  0.40  0.57  1.00
anxious   -0.32 -0.37 -0.05 -0.16 -0.34  1.00
jittery   -0.34 -0.41 -0.06 -0.14 -0.40  0.56  1.00
nervous   -0.38 -0.40 -0.15 -0.24 -0.38  0.58  0.58  1.00
tense     -0.47 -0.49 -0.17 -0.31 -0.47  0.57  0.54  0.63  1.00
upset     -0.35 -0.30 -0.25 -0.35 -0.32  0.32  0.22  0.39  0.45  1.00
```

Get the msqR items and subjects that correspond with sai

R code

```
dim(msqR)
 table(msqR['time'])
 msqR.once <- msqR[msqR['time']==1,]
dim(msqR.once)
 lowerCor(msqR.once[msq.items])
```

```
 dim(msqR)
[1] 6411   88
>  table(msqR['time'])
time
   1    2    3    4
3032 2086 1112  181
>  msqR.once <- msqR[msqR['time']==1,]
> dim(msqR.once)
[1] 3032   88
> lowerCor(msqR.once[msq.items])
          at.es calm  cnfdn cntnt relxd anxis jttry nervs tense upset
at.ease    1.00
calm       0.62  1.00
confident  0.44  0.31  1.00
content    0.56  0.45  0.61  1.00
relaxed    0.60  0.56  0.34  0.45  1.00
anxious   -0.23 -0.28  0.01 -0.07 -0.27  1.00
jittery   -0.24 -0.32  0.02 -0.04 -0.34  0.47  1.00
nervous   -0.29 -0.31 -0.08 -0.13 -0.30  0.54  0.48  1.00
tense     -0.35 -0.36 -0.07 -0.19 -0.40  0.57  0.48  0.57  1.00
upset     -0.30 -0.23 -0.17 -0.29 -0.29  0.29  0.16  0.35  0.45  1.00
```

Compare the two correlation matrices

R code

```
R.sai <-lowerCor(sai.once[msq.items])
R.msqR <- lowerCor(msqR.once[msq.items])
R.lowup <- lowerUpper(lower=R.sai,upper=R.msqR, diff=TRUE)
corPlot(R.lowup,main = "Similarity of SAI correlations and MSQR correl
round(R.lowup,2)   #sai below the diagonal  sai - msq above the diagon
```

```
round(R.lowup,2)   #sai below the diagonal  (sai - msq) above the diagonal
           at.ease  calm confident content relaxed anxious jittery nervous tense upset
at.ease         NA  0.07      0.02    0.02    0.09   -0.09   -0.10   -0.09 -0.12 -0.05
calm          0.69    NA      0.02    0.03    0.12   -0.09   -0.10   -0.09 -0.12 -0.07
confident     0.45  0.34        NA   -0.07    0.05   -0.06   -0.08   -0.08 -0.11 -0.08
content       0.58  0.48      0.54      NA    0.12   -0.09   -0.10   -0.11 -0.12 -0.07
relaxed       0.69  0.68      0.40    0.57      NA   -0.07   -0.06   -0.09 -0.08 -0.04
anxious      -0.32 -0.37     -0.05   -0.16   -0.34      NA    0.09    0.03  0.01  0.03
jittery      -0.34 -0.41     -0.06   -0.14   -0.40    0.56      NA    0.10  0.06  0.07
nervous      -0.38 -0.40     -0.15   -0.24   -0.38    0.58    0.58      NA  0.06  0.03
tense        -0.47 -0.49     -0.17   -0.31   -0.47    0.57    0.54    0.63    NA  0.00
upset        -0.35 -0.30     -0.25   -0.35   -0.32    0.32    0.22    0.39  0.45    NA
```

We want to find scales (and their scores) from the `sai.once` and `msqR.once` data

1. We use the scoring keys for the overlapping `sai` and `msqR` items.
2. The `scoreitems` function will find the scores and give us some statistics.
3. The `scoreOverlap` function corrects for overlapping items and also gives statistics. It will not give scores.
4. We then will want to compare these scores across the sai and msqR

score the sai

R code

```
sai.scales <- scoreItems(sai.msq.keys, sai.once)
 sai.scales
```

```
Call: scoreItems(keys = sai.msq.keys, items = sai.once)
(Unstandardized) Alpha:
       pos  neg  anx
alpha 0.85 0.82 0.87
Standard errors of unstandardized Alpha:
         pos   neg    anx
ASE   0.0098 0.011 0.0062
Average item correlation:
           pos  neg  anx
average.r 0.54 0.48 0.39
Median item correlation:
 pos  neg  anx
0.55 0.55 0.38

 Guttman 6* reliability:
          pos  neg  anx
Lambda.6 0.85 0.82 0.89
Signal/Noise based upon av.r :
             pos neg anx
Signal/Noise 5.8 4.7 6.5

Scale intercorrelations corrected for attenuation
 raw correlations below the diagonal, alpha on the diagonal
 corrected correlations above the diagonal:
      pos   neg   anx
pos  0.85 -0.59 -1.03
neg -0.50  0.82  0.99
anx -0.89  0.84  0.87
```

Do this for the msqR.items

R code

```
msq.scales <- scoreItems(sai.msq.keys, msqR.once)
msq.scales
```

```
all: scoreItems(keys = sai.msq.keys, items = msqR.once)
(Unstandardized) Alpha:
       pos  neg  anx
alpha 0.83 0.76 0.82
Standard errors of unstandardized Alpha:
       pos   neg    anx
ASE   0.01 0.012 0.0074
Average item correlation:
           pos  neg  anx
average.r 0.49 0.39 0.32
Median item correlation:
 pos  neg  anx
0.50 0.43 0.31
 Guttman 6* reliability:
          pos  neg  anx
Lambda.6 0.82 0.75 0.85
Signal/Noise based upon av.r :
             pos neg anx
Signal/Noise 4.9 3.2 4.6

Scale intercorrelations corrected for attenuation
 raw correlations below the diagonal, alpha on the diagonal
 corrected correlations above the diagonal:
      pos   neg   anx
pos  0.83 -0.48 -1.07
neg -0.38  0.76  0.98
anx -0.88  0.78  0.82
```

Compare them

R code

```
names(msq.scales)
sai.scores <- sai.scales$scores
msq.scores <- msq.scales$scores
sai.msq <- data.frame(sai=sai.scores, msq=msq.scores)
lowerCor(sai.msq)
```

```
names(msq.scales)
 [1] "scores"          "missing"        "alpha"          "av.r"           "sn"
 [6] "n.items"         "item.cor"       "cor"            "corrected"      "G6"
[11] "item.corrected" "response.freq"  "raw"            "ase"            "med.r"
[16] "keys"            "Call"   ]

lowerCor(sai.msq)
        sa.ps sa.ng sa.nx msq.p msq.ng msq.nx
sai.pos  1.00
sai.neg -0.50  1.00
sai.anx -0.89  0.84  1.00
msq.pos  0.83 -0.42 -0.74  1.00
msq.neg -0.45  0.84  0.72 -0.38  1.00
msq.anx -0.80  0.72  0.88 -0.88  0.78   1.00
```

Let's make a table to show the comparisons

R code

```
alternate <- diag(cor2(sai.scores,msq.scores))
stats.df <- data.frame(sai = t(sai.scales$alpha), msq=t(msq.scales$alp
 colnames(stats.df ) <- c("sai alpha", "msq alpha", "alternate form")
 round(stats.df,2)
```

```
alternate <- diag(cor2(sai.scores,msq.scores))
      pos   neg   anx
pos  0.83 -0.45 -0.80
neg -0.42  0.84  0.72
anx -0.74  0.72  0.88
> stats.df <- data.frame(sai = t(sai.scales$alpha), msq=t(msq.scales$alpha), alternate  = alt
>  colnames(stats.df ) <- c("sai alpha", "msq alpha", "alternate form")
>  round(stats.df,2)
    sai alpha msq alpha alternate form
pos      0.85      0.83           0.83
neg      0.82      0.76           0.84
anx      0.87      0.82           0.88
```

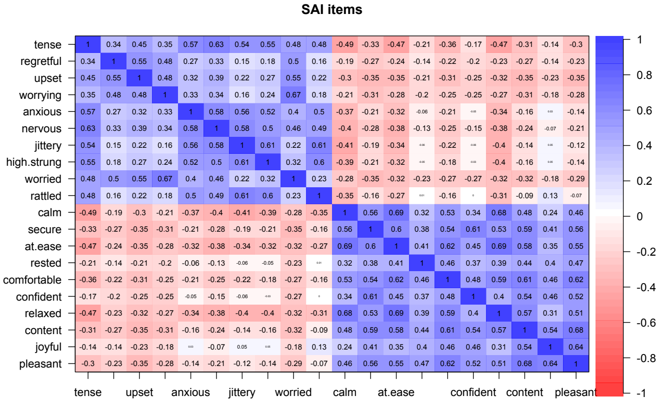
More examples

1. The Week 3a RMD file has more examples.
2. Using the ICAR-16 item set from the `ability`data set.
   - Open source ability items developed by Condon and Revelle (2014) and further discussed by (Revelle et al., 2020) and (Dworak et al., 2021).
3. Compare alternative splits of the iCAR (odd versus even is high, first versus last is lower)
4. using a *keys* list to define direction of keying for items
5. Examples from the `bfi` and `spi` data sets.
6. See the “how to" use R for scoring scales.

Is a scale really one scale?

1. α tells us what the average correlation is (times n .items)
2. $alpha = \frac{k\bar{r}}{1+(k-1)\bar{r}}$
3. Consider all 20 items from the sai
4. `corPlot(sai.once[selectFromKeys(sai.keys[1])], main="SAI items")`
5. `alpha(sai.once[selectFromKeys(sai.keys[1])], check.keys=TRUE)`
6. What is the factor structure?

sai items


SAI items
tense 1 0.34 0.45 0.35 0.57 0.63 0.54 0.55 0.48 0.48 -0.49 -0.33 -0.47 -0.21 -0.36 -0.17 -0.47 -0.31 -0.14 -0.3
regretful 0.34 1 0.55 0.48 0.27 0.33 0.15 0.18 0.5 0.16 -0.19 -0.27 -0.24 -0.14 -0.22 -0.2 -0.23 -0.27 -0.14 -0.23
upset 0.45 0.55 1 0.48 0.32 0.39 0.22 0.27 0.55 0.22 -0.3 -0.35 -0.35 -0.21 -0.31 -0.25 -0.32 -0.35 -0.23 -0.35
worrying 0.35 0.48 0.48 1 0.33 0.34 0.16 0.24 0.67 0.18 -0.21 -0.31 -0.28 -0.2 -0.25 -0.25 -0.27 -0.31 -0.18 -0.28
anxious 0.57 0.27 0.32 0.33 1 0.58 0.56 0.52 0.4 0.5 -0.37 -0.21 -0.32 -0.06 -0.21 -0.34 -0.16 -0.14
nervous 0.63 0.33 0.39 0.34 0.58 1 0.58 0.5 0.46 0.49 -0.4 -0.28 -0.38 -0.13 -0.25 -0.15 -0.38 -0.24 -0.07 -0.21
jittery 0.54 0.15 0.22 0.16 0.56 0.58 1 0.61 0.22 0.61 -0.41 -0.19 -0.34 -0.22 -0.4 -0.14 -0.12
high.strung 0.55 0.18 0.27 0.24 0.52 0.5 0.61 1 0.32 0.6 -0.39 -0.21 -0.32 -0.18 -0.4 -0.16 -0.14
worried 0.48 0.5 0.55 0.67 0.4 0.46 0.22 0.32 1 0.23 -0.28 -0.35 -0.32 -0.23 -0.27 -0.27 -0.32 -0.32 -0.18 -0.29
rattled 0.48 0.16 0.22 0.18 0.5 0.49 0.61 0.6 0.23 1 -0.35 -0.16 -0.27 -0.16 0 -0.31 -0.09 0.13 -0.07
calm -0.49 -0.19 -0.3 -0.21 -0.37 -0.4 -0.41 -0.39 -0.28 -0.35 1 0.56 0.69 0.32 0.53 0.34 0.68 0.48 0.24 0.46
secure -0.33 -0.27 -0.35 -0.31 -0.21 -0.28 -0.19 -0.21 -0.35 -0.16 0.56 1 0.6 0.38 0.54 0.61 0.53 0.59 0.41 0.56
at.ease -0.47 -0.24 -0.35 -0.28 -0.32 -0.38 -0.34 -0.32 -0.32 -0.27 0.69 0.6 1 0.41 0.62 0.45 0.69 0.58 0.35 0.55
rested -0.21 -0.14 -0.21 -0.2 -0.06 -0.13 -0.06 -0.05 -0.23 0.32 0.38 0.41 1 0.46 0.37 0.39 0.44 0.4 0.47
comfortable -0.36 -0.22 -0.31 -0.25 -0.21 -0.25 -0.22 -0.18 -0.27 -0.16 0.53 0.54 0.62 0.46 1 0.48 0.59 0.61 0.46 0.62
confident -0.17 -0.2 -0.25 -0.25 -0.05 -0.15 -0.06 -0.27 0 0.34 0.61 0.45 0.37 0.48 1 0.4 0.54 0.46 0.52
relaxed -0.47 -0.23 -0.32 -0.27 -0.34 -0.38 -0.4 -0.4 -0.32 -0.31 0.68 0.53 0.69 0.39 0.59 0.4 1 0.57 0.31 0.51
content -0.31 -0.27 -0.35 -0.31 -0.16 -0.24 -0.14 -0.16 -0.32 -0.09 0.48 0.59 0.58 0.44 0.61 0.54 0.57 1 0.54 0.68
joyful -0.14 -0.14 -0.23 -0.18 -0.07 0.05 -0.18 0.13 0.24 0.41 0.35 0.4 0.46 0.46 0.31 0.54 1 0.64
pleasant -0.3 -0.23 -0.35 -0.28 -0.14 -0.21 -0.12 -0.14 -0.29 -0.07 0.46 0.56 0.55 0.47 0.62 0.52 0.51 0.68 0.64 1
tense upset anxious jittery worried calm at.ease confident content pleasant
1 0.8 0.6 0.4 0.2 0 -0.2 -0.4 -0.6 -0.8 -1

What is the factor structure of the sai?

R code

```
nfactors(sai.once[4:23])
f1 <- fa(sai.once[4:23])
f2 <- fa(sai.once[4:23],2)
summary(f1)
 summary(f2)
```

```
Number of factors
Call: vss(x = x, n = n, rotate = rotate, diagonal = diagonal, fm = fm,
    n.obs = n.obs, plot = FALSE, title = title, use = use, cor = cor)
VSS complexity 1 achieves a maximimum of 0.77  with  1  factors
VSS complexity 2 achieves a maximimum of 0.9  with  2  factors
The Velicer MAP achieves a minimum of 0.01  with  3  factors
Empirical BIC achieves a minimum of  -517.33  with  6  factors
Sample Size adjusted BIC achieves a minimum of  -122.77  with  9  factors
```

factoring the sai

R code

```
f1 <- fa(sai.once[4:23])
f2 <- fa(sai.once[4:23],2)
summary(f1)
 summary(f2)
```

```
Factor analysis with Call: fa(r = sai.once[4:23])

Test of the hypothesis that 1 factor is sufficient.
The degrees of freedom for the model is 170  and the objective function was  4.67
The number of observations was  3032  with Chi Square =  14131.73  with prob <  0
The root mean square of the residuals (RMSA) is  0.15
The df corrected root mean square of the residuals is  0.16
Tucker Lewis Index of factoring reliability =  0.538
RMSEA index =  0.165  and the 10 % confidence intervals are  0.162 0.167
BIC =  12768.85

>  summary(f2)
Factor analysis with Call: fa(r = sai.once[4:23], nfactors = 2)
Test of the hypothesis that 2 factors are sufficient.
The degrees of freedom for the model is 151  and the objective function was  1.81
The number of observations was  3032  with Chi Square =  5466.88  with prob <  0

The root mean square of the residuals (RMSA) is  0.07
The df corrected root mean square of the residuals is  0.08

Tucker Lewis Index of factoring reliability =  0.802
RMSEA index =  0.108  and the 10 % confidence intervals are  0.105 0.11
BIC =  4256.31
 With factor correlations of
      MR1   MR2
MR1  1.00 -0.32
MR2 -0.32  1.00
```

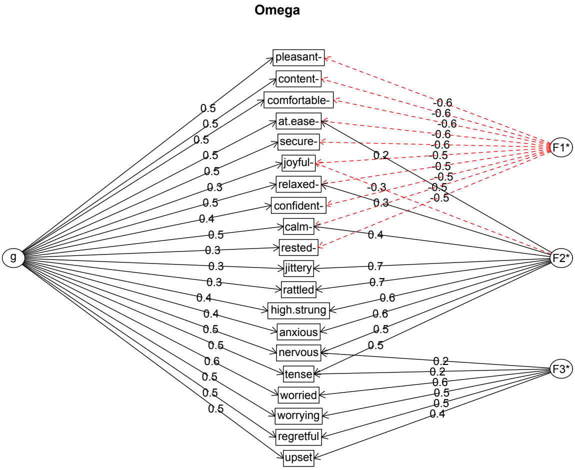
By default, omega gives a 3 factor solution

R code

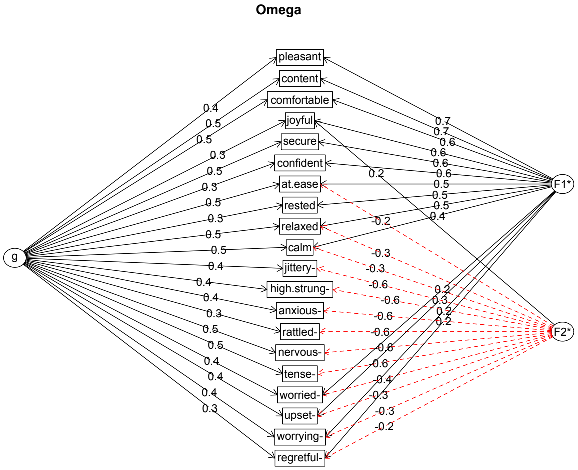
```
om <- omega(sai.once[4:23])
om2 <- omega(sai.once[4:23],2) #but can do fewer
summary(om)
```

```
Omega
omega(m = sai.once[4:23])
Alpha:                  0.91
G.6:                    0.94
Omega Hierarchical:     0.55
Omega H asymptotic:     0.58
Omega Total             0.94

With eigenvalues of:
  g F1* F2* F3*
4.2 3.2 2.7 1.2
The degrees of freedom for the model is 133  and the fit was  0.66
The number of observations was  3032  with Chi Square =  1996.43  with prob <  0

The root mean square of the residuals is  0.03
The df corrected root mean square of the residuals is  0.03

RMSEA and the  0.9 confidence intervals are  0.068 0.065 0.071
BIC =  930.17Explained Common Variance of the general factor =  0.37

 Total, General and Subset omega for each subset
                                                g  F1*  F2*  F3*
Omega total for total scores and subscales    0.94 0.92 0.86 0.81
Omega general for total scores and subscales  0.55 0.35 0.27 0.43
Omega group for total scores and subscales    0.33 0.57 0.59 0.39
```

omega for sai.once



omega for sai.once



omega with 2 factors

R code

```
summary(om2)
```

```
Omega
omega(m = sai.once[4:23], nfactors = 2)
Alpha:                 0.91
G.6:                   0.94
Omega Hierarchical:    0.45
Omega H asymptotic:    0.48
Omega Total            0.93

With eigenvalues of:
  g F1* F2*
3.4 3.5 2.9
The degrees of freedom for the model is 151  and the fit was  1.81
The number of observations was  3032  with Chi Square =  5466.88  with prob <  0

The root mean square of the residuals is  0.07
The df corrected root mean square of the residuals is  0.08

RMSEA and the  0.9 confidence intervals are  0.108 0.105 0.11
BIC =  4256.31Explained Common Variance of the general factor =  0.35

 Total, General and Subset omega for each subset
                                                 g  F1*  F2*
Omega total for total scores and subscales    0.93 0.90 0.83
Omega general for total scores and subscales  0.45 0.32 0.33
Omega group for total scores and subscales    0.37 0.58 0.49
```

Internal consistency estimates of the `sai`

1. Although a simple `alpha` analysis would suggest that the `sai` has high internal consistency and that 91% of the variance is reliable variance.
2. But the `omega` suggests that only 45% of the total score reflects one thing (that is to say, what ever it is measuring less than half of its variance is one thing)
3. We see this also when we look at the split half values or use the `reliability` function.

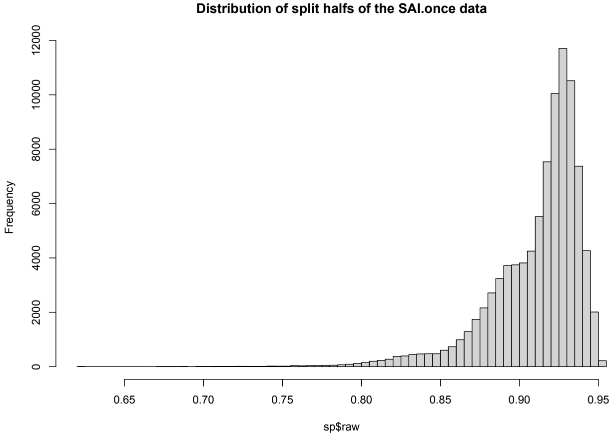
Split half and reliability functions

R code

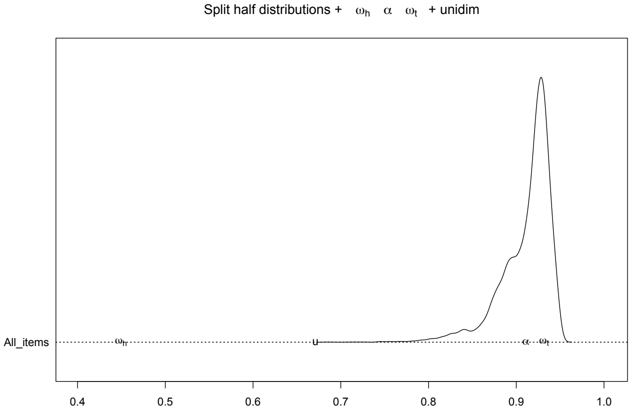
```
round(choose(20,10))/2   #how many splits to try (defaults to 10,000)
rel <- reliability(sai.once[4:23])
plot(rel)
sp <- splitHalf(sai.once[4:23],raw=TRUE,n.sample=92378)
hist(sp$raw,breaks=51,main="Distribution of split halfs of the SAI.onc
sp
```

```
round(choose(20,10))/2
[1] 92378


Measures of reliability
reliability(keys = sai.once[4:23])
          omega_h alpha omega.tot  Uni r.fit fa.fit max.split min.split mean.r med.r n.items
All_items    0.45  0.91      0.93 0.67  0.79   0.85      0.95      0.68   0.34  0.32      20

Split half reliabilities
Call: splitHalf(r = sai.once[4:23], raw = TRUE, n.sample = 92378)

Maximum split half reliability (lambda 4) =  0.95
Guttman lambda 6                          =  0.94
Average split half reliability            =  0.91
Guttman lambda 3 (alpha)                  =  0.91
Guttman lambda 2                          =  0.92
Minimum split half reliability  (beta)    =  0.62
Average interitem r =  0.34  with median =  0.32
                                             2.5% 50% 97.5%
 Quantiles of split half reliability      =  0.83 0.92 0.94
```

Plotting the distributions



Plotting the distributions



An alternative to classical test theory is item response theory

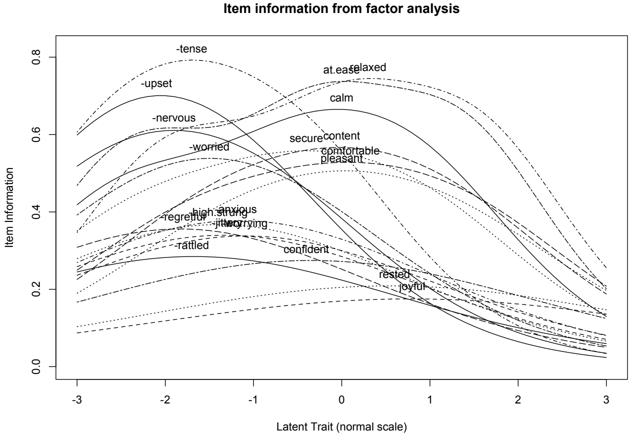
1. Classical theory treats items as random replicates of each other
2. IRT tries to model the item and person response.
3. Although seemingly very different, the two approaches may be combined with non-linear factor analysis.
4. This treats the item responses in terms of their tetrachoric or polychoric correlations.
5. the `irt.fa` will do this

R code

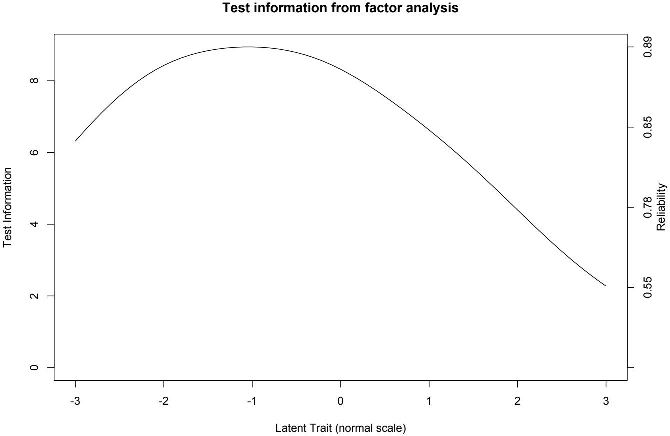
```
f1.irt <- irt.fa(sai.once[4:23])
f1.irt
```

```
Item Response Analysis using Factor Analysis

Call: irt.fa(x = sai.once[4:23])
Item Response Analysis using Factor Analysis

 Summary information by factor and item
 Factor =  1
               -3   -2   -1    0    1    2    3
calm         0.42 0.53 0.61 0.67 0.57 0.33 0.13
secure       0.35 0.48 0.54 0.55 0.46 0.29 0.13
tense        0.61 0.78 0.75 0.56 0.28 0.11 0.03
regretful    0.31 0.35 0.33 0.25 0.16 0.09 0.05
at.ease      0.47 0.62 0.65 0.74 0.70 0.51 0.20
upset        0.60 0.70 0.58 0.35 0.16 0.06 0.02
worrying     0.24 0.31 0.34 0.30 0.22 0.14 0.08
rested       0.10 0.14 0.18 0.21 0.21 0.18 0.15
anxious      0.27 0.35 0.38 0.33 0.24 0.15 0.08
comfortable  0.25 0.40 0.49 0.53 0.49 0.37 0.21
confident    0.17 0.23 0.27 0.27 0.24 0.18 0.12
nervous      0.52 0.61 0.55 0.38 0.20 0.09 0.03
jittery      0.26 0.32 0.34 0.28 0.20 0.13 0.07
high.strung  0.28 0.35 0.36 0.30 0.20 0.12 0.07
relaxed      0.35 0.59 0.65 0.74 0.72 0.56 0.26
content      0.23 0.39 0.52 0.57 0.51 0.36 0.19
worried      0.39 0.52 0.52 0.40 0.24 0.12 0.05
rattled      0.24 0.28 0.27 0.22 0.16 0.10 0.06
joyful       0.09 0.12 0.15 0.17 0.17 0.16 0.14
pleasant     0.19 0.33 0.46 0.51 0.46 0.34 0.20
Test Info    6.32 8.43 8.94 8.32 6.63 4.40 2.27
SEM          0.40 0.34 0.33 0.35 0.39 0.48 0.66
Reliability  0.84 0.88 0.89 0.88 0.85 0.77 0.56
```

Plotting the information function of each item



Plotting the information function of the test



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

Dworak, E. M., Revelle, W., Doebler, P., and Condon, D. M. (2021). Using the International Cognitive Ability Resource as an open source tool to explore individual differences in cognitive ability. *Personality and Individual Differences*, 169.

McNemar, Q. (1946). Opinion-attitude methodology. *Psychological Bulletin*, 43(4):289–374.

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

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

Spearman, C. (1904). The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101.