

Cronbach's Alpha

Joseph Nathan Cohen

Fall 2019

Contents

Explanation	1
Model	1
Implementation	1
Data	1
Step One: Calculate a Correlation Matrix	2
Step Two: Calculate Cronbach's Alpha	3
Step Three: Interpret the Results	4

Explanation

Cronbach's Alpha tests of the assumption that a set of variables reliably predict a common underlying concept. It tests whether there are strong relationships among variables that purport to measure the same underlying construct. If this tests fails, then you might question whether that variable set does in fact measure the same thing.

Model

We can calculate Cronbach's Alpha via our variab'es correlations, where:¹

$$\alpha = \frac{N \times \bar{c}}{v + (N - 1) \cdot \bar{c}}$$

Where:

- α is the Cronbach's Alpha score
- N is the number of variables being tested
- \bar{c} is the average of covariance between variables
- \bar{v} is the average variance of each variable

Roughly, the metric approximates the ratio of inter-item covariance to our variables' overall variance.

Implementation

Data

Our data is this module's simulated emotional disposition data, contained in the Excel spreadsheet "Simulated Emotional Disposition Data for EFA.xlsx". The set scores 400 respondents over 15 traits: happiness, optimism, sociability, anxiety, anger, jealousy, resentment, fear, boredom, tiredness, annoyance, irritability, hopefulness, friendliness, and ambition. These traits are all rated on a zero (inapplicable) to ten (fully applicable).

¹following Goforth, Chelsea (2015) "Using and Interpreting Chronbach's Alpha" University of Virginia Library Research Data Services <https://data.library.virginia.edu/using-and-interpreting-cronbachs-alpha/>

```

library(readxl)
DATA <- read_xlsx("Simulated Emotional Disposition Data.xlsx", sheet = 1)
DATA <- data.frame(DATA)

#Rounding the data's scores to make save space in these printouts:
DATA[2:16] <- round(DATA[2:16], 2)

#A peek at the first few data points in the set:
head(DATA[1:7], 5)

```

```

##   id happiness optimism sociability anxiety anger jealousy
## 1  1     5.64     6.66         7.45    5.12  5.30     4.58
## 2  2     6.73     7.71         6.73    3.29  7.84     6.48
## 3  3     4.74     5.16         5.82    2.64  4.88     7.13
## 4  4     3.32     3.11         2.93    4.73  3.51     4.56
## 5  5     5.72     6.00         7.83    6.80  6.05     5.96

```

Step One: Calculate a Correlation Matrix

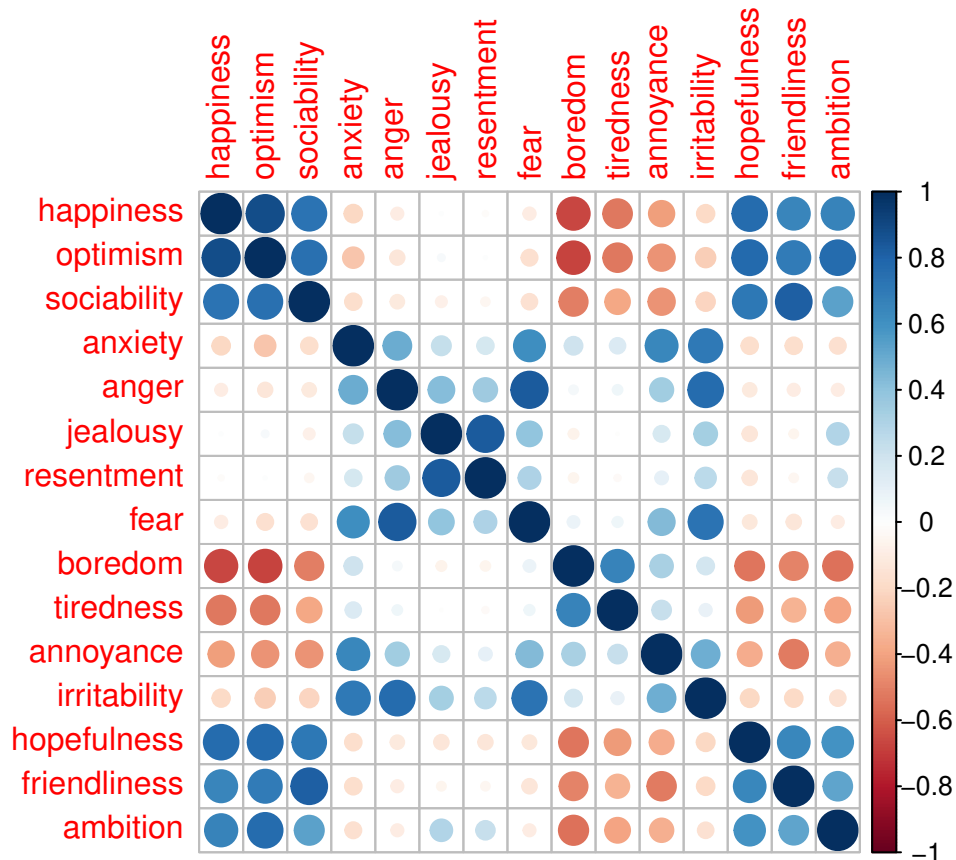
Our first step is to create a matrix of correlations among the items we are trying to index. If our data is continuous, we can calculate these correlations using the `cor()` command in the base package. If you are working with binary or ordinal data, use the `polychoric()` command in the *psych* package.

```

#We are using variables 2 - 16 in this data set.
#They are all continuous, so we use cor().
#Let's start off by finding highly related metrics:
CORRS <- cor(DATA[2:16])

#Visualizing using corrplot (see next page)
library(corrplot)
corrplot(CORRS)

```



The results suggest a very strong relationship between happiness, optimism, sociability, hopefulness, friendliness, and ambition. We construct a correlation matrix with only these variables:

```
#Calculate correlation matrix with variables we are going to index
CORRS.2 <- cor(DATA[c(2:4,14:16)])

#A peek at the top of the matrix:
round(CORRS.2, 2)[,1:5]
```

```
##           happiness optimism sociability hopefulness friendliness
## happiness      1.00    0.88      0.74      0.76      0.65
## optimism       0.88    1.00      0.74      0.77      0.70
## sociability    0.74    0.74      1.00      0.72      0.81
## hopefulness    0.76    0.77      0.72      1.00      0.64
## friendliness   0.65    0.70      0.81      0.64      1.00
## ambition       0.66    0.76      0.54      0.59      0.52
```

Remember that the variables you are testing should have positive correlations. If this happens, reverse-code and reconceptualize the variable you are testing. For example, *happiness* and *boredom* share a strong, negative relationship. If I wanted to measure their association in a Cronbach's Alpha test, I might do something like reverse-code *boredom* and call it something like *engagedness*.

Step Two: Calculate Cronbach's Alpha

We calculate a Cronbach's Alpha using the `alpha()` command in the *psych* package:

```
library(psych)
alpha <- alpha(CORRS.2)
```

#Results:

alpha

```
##
## Reliability analysis
## Call: alpha(x = CORRS.2)
##
##   raw_alpha std.alpha G6(smc) average_r S/N median_r
##     0.93     0.93   0.94     0.7  14     0.72
##
## Reliability if an item is dropped:
##           raw_alpha std.alpha G6(smc) average_r S/N var.r med.r
## happiness      0.91     0.91   0.92     0.68 10.6 0.0103 0.71
## optimism       0.91     0.91   0.90     0.66  9.9 0.0090 0.66
## sociability    0.92     0.92   0.91     0.70 11.4 0.0107 0.68
## hopefulness    0.92     0.92   0.93     0.70 11.7 0.0126 0.72
## friendliness   0.93     0.93   0.92     0.72 12.7 0.0096 0.74
## ambition       0.93     0.93   0.93     0.74 14.4 0.0052 0.74
##
## Item statistics
##           r r.cor r.drop
## happiness  0.90 0.89 0.86
## optimism   0.93 0.94 0.90
## sociability 0.87 0.85 0.82
## hopefulness 0.86 0.82 0.80
## friendliness 0.83 0.80 0.76
## ambition   0.79 0.73 0.69
```

Step Three: Interpret the Results

Key elements of the output:

- **raw_alpha** return above gives Cronbach's Alpha based on covariances.
- **std.alpha** is a standardized alpha based on correlations.
- **average_r** is the average correlation between the variables
- **median_r** is the median correlation among these variables

The lower panel of the output gives you a sense of how our results would change if we were to drop one of the items in our variable set.

Focus on **raw_alpha**. In this case, the score is 0.93.

- A score about 0.80 is conventionally treated as sufficient in the peer-reviewed literature.
- Some methodologists argue that the threshold could be lowered to 0.70, though doing so injects anticonservatism into your analysis.
- I have also heard the argument that an alpha over ~0.95 could be a sign that your analysis is relying on redundant variables, rather than multiple metrics that triangulate the measurement of a common underlying construct.