Coefficient α (Part II)

```r
library(psych)
library(tidyverse)
library(modelsummary)  # for summarizing data
```

```
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
```

```
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
```

We'll once again use the `bfi` data set, this time with the Openness items

```
data(bfi, package = "psych")
head(bfi[c("O1", "O2", "O3", "O4", "O5")])
```

```
      O1 O2 O3 O4 O5
61617  3  6  3  4  3
61618  4  2  4  3  3
61620  4  2  5  5  2
61621  3  3  4  3  5
61622  3  3  4  3  3
61623  4  3  5  6  1
```

Descriptive Statistics

```
# Get means and variances
mv <- datasummary(O1 + O2 + O3 + O4 + O5 ~ Mean + Var, data = bfi,
                  output = "data.frame")
# Correlation, adding means and variances
datasummary_correlation(bfi[c("O1", "O2", "O3", "O4", "O5")],
                        add_columns = mv[-1])
```

```
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
```

```
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
```

	O1	O2	O3	O4	O5	Mean	Var
O1	1	.	.	.	.	4.82	1.28
O2	−0.21	1	.	.	.	2.71	2.45
O3	0.40	−0.26	1	.	.	4.44	1.49
O4	0.18	−0.07	0.19	1	.	4.89	1.49
O5	−0.24	0.32	−0.31	−0.18	1	2.49	1.76

```
See help("Deprecated")
Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")

Warning in attr(x, "align"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")

Warning in attr(x, "format"): 'xfun::attr()' is deprecated.
Use 'xfun::attr2()' instead.
See help("Deprecated")
```

You can find that O2 and O5 have negative correlations with other items

α Without Recoding

```r
cov1 <- cov(bfi[c("O1", "O2", "O3", "O4", "O5")], use = "complete")
# Compute alpha by hand using formula
5 / (5 - 1) * (1 - sum(diag(cov1)) / sum(cov1))
```

```
[1] -0.1568749
```

```r
# Using `psych::alpha()`; the estimate is labelled "raw_alpha"
psych::alpha(bfi[c("O1", "O2", "O3", "O4", "O5")])
```

```
Warning in psych::alpha(bfi[c("O1", "O2", "O3", "O4", "O5")]): Some items were negatively co
should be reversed.
To do this, run the function again with the 'check.keys=TRUE' option
```

```
Some items ( O2 O5 ) were negatively correlated with the first principal component and
probably should be reversed.
To do this, run the function again with the 'check.keys=TRUE' option


Reliability analysis
Call: psych::alpha(x = bfi[c("O1", "O2", "O3", "O4", "O5")])

  raw_alpha std.alpha G6(smc) average_r    S/N   ase mean   sd median_r
     -0.13    -0.097   0.096    -0.018 -0.088 0.035  3.9 0.56    -0.12

    95% confidence boundaries
         lower alpha upper
Feldt     -0.2 -0.13 -0.07
Duhachek  -0.2 -0.13 -0.06

 Reliability if an item is dropped:
   raw_alpha std.alpha G6(smc) average_r    S/N alpha se var.r    med.r
O1    -0.194    -0.235 -0.0276   -0.0500 -0.191    0.037 0.066 -0.12369
O2    -0.031     0.026  0.1617    0.0066  0.027    0.032 0.082 -0.00034
O3    -0.096    -0.146  0.0089   -0.0328 -0.127    0.033 0.054 -0.12369
O4    -0.232    -0.240  0.0297   -0.0508 -0.193    0.039 0.103 -0.22636
O5     0.028     0.135  0.2170    0.0374  0.156    0.031 0.067  0.05503

 Item statistics
      n raw.r std.r r.cor r.drop mean  sd
O1 2778  0.42  0.52  0.46  0.020  4.8 1.1
O2 2800  0.49  0.36 -0.11 -0.088  2.7 1.6
O3 2772  0.39  0.47  0.37 -0.054  4.4 1.2
O4 2786  0.48  0.52  0.28  0.039  4.9 1.2
O5 2780  0.36  0.28 -0.33 -0.136  2.5 1.3

Non missing response frequency for each item
      1    2    3    4    5    6 miss
O1 0.01 0.04 0.08 0.22 0.33 0.33 0.01
O2 0.29 0.26 0.14 0.16 0.10 0.06 0.00
O3 0.03 0.05 0.11 0.28 0.34 0.20 0.01
O4 0.02 0.04 0.06 0.17 0.32 0.39 0.01
O5 0.27 0.32 0.19 0.13 0.07 0.03 0.01
```

You can see that α is negative.

Recode

```
bfi$O2r <- 7 - bfi$O2
bfi$O5r <- 7 - bfi$O5
```

α After Recoding

```
cov2 <- cov(bfi[c("O1", "O2r", "O3", "O4", "O5r")], use = "complete")
# Compute alpha by hand using formula
5 / (5 - 1) * (1 - sum(diag(cov2)) / sum(cov2))
```

```
[1] 0.6025464
```

```
# Using `psych::alpha()`; the estimate is labelled "raw_alpha"
psych::alpha(bfi[c("O1", "O2r", "O3", "O4", "O5r")])
```

```
Reliability analysis
Call: psych::alpha(x = bfi[c("O1", "O2r", "O3", "O4", "O5r")])

  raw_alpha std.alpha G6(smc) average_r S/N   ase mean   sd median_r
       0.6      0.61    0.57      0.24 1.5 0.012  4.6 0.81     0.23

    95% confidence boundaries
         lower alpha upper
Feldt     0.58   0.6  0.62
Duhachek  0.58   0.6  0.62

 Reliability if an item is dropped:
    raw_alpha std.alpha G6(smc) average_r S/N alpha se  var.r med.r
O1       0.53      0.53    0.48      0.22 1.1    0.014 0.0092  0.23
O2r      0.57      0.57    0.51      0.25 1.3    0.013 0.0076  0.22
O3       0.50      0.50    0.44      0.20 1.0    0.015 0.0071  0.20
O4       0.61      0.62    0.56      0.29 1.6    0.012 0.0044  0.29
O5r      0.51      0.53    0.47      0.22 1.1    0.015 0.0116  0.20

 Item statistics
       n raw.r std.r r.cor r.drop mean  sd
```

```
O1  2778  0.62  0.65  0.52   0.39  4.8 1.1
O2r 2800  0.65  0.60  0.43   0.33  4.3 1.6
O3  2772  0.67  0.69  0.59   0.45  4.4 1.2
O4  2786  0.50  0.52  0.29   0.22  4.9 1.2
O5r 2780  0.67  0.66  0.52   0.42  4.5 1.3

Non missing response frequency for each item
       1    2    3    4    5    6 miss
O1  0.01 0.04 0.08 0.22 0.33 0.33 0.01
O2r 0.06 0.10 0.16 0.14 0.26 0.29 0.00
O3  0.03 0.05 0.11 0.28 0.34 0.20 0.01
O4  0.02 0.04 0.06 0.17 0.32 0.39 0.01
O5r 0.03 0.07 0.13 0.19 0.32 0.27 0.01
```

Here is an example for reporting α in a manuscript:

> **Tip**
>
> *The reliability coefficient of the composite scores of the five Openness items for our sample was .60, 95% CI [.58, .62].*

Multilevel Reliability

```r
# Example data from Bolger & Laurenceau (2013) chapter 7
dat <- read.csv(
    "https://raw.githubusercontent.com/The-Change-Lab/collaborations/refs/heads/main/Bolger_
    header = TRUE
)
# Reshape the data to have items in multiple columns, which is more common
dat2 <- dat |>
    pivot_wider(names_from = "item", names_prefix = "x",
                values_from = "y")
head(dat2, 20)
```

```
# A tibble: 20 x 6
   person  time    x1    x2    x3    x4
    <int> <int> <int> <int> <int> <int>
 1    301     1     2     2     3     4
 2    301     2     2     3     3     2
 3    301     3     3     3     2     2
```

```
 4    301     4     4     3     3     3
 5    301     5     2     2     2     2
 6    301     6     2     2     1     2
 7    301     7     2     2     1     2
 8    301     8     2     2     2     2
 9    301     9     2     3     2     2
10    301    10     3     3     3     3
11    309     1     3     3    NA     1
12    309     2     3     3     3     3
13    309     3     3     3     3     1
14    309     4     3     1     3     1
15    309     5     1     1     1     1
16    309     6     3     1     1     1
17    309     7     1     1     3     1
18    309     8     1     1     1     1
19    309     9     1     1     1     1
20    309    10     1     1     1     1
```

As can be seen, each person has 10 time points, each time with 4 items. We can get three types of composites:

```r
# Raw composite
xsum <- rowSums(dat2[c("x1", "x2", "x3", "x4")])
var(xsum, na.rm = TRUE)  # total variance
```

```
[1] 11.88269
```

```r
# Between composite (one for each person)
xsumb <- tapply(xsum, INDEX = dat2$person, FUN = mean)
var(xsumb, na.rm = TRUE)  # between variance
```

```
[1] 7.962253
```

```r
# Within composite
xsumw <- xsum - ave(xsum, dat2$person)
var(xsumw, na.rm = TRUE)  # within variance
```

```
[1] 3.91179
```

And you can compute reliability for each composite using the following code:

```r
source("https://github.com/marklhc/mcfa_reliability_supp/raw/master/multilevel_alpha.R")
multilevel_alpha(dat2[c("x1", "x2", "x3", "x4")],
                 id = dat2$person)
```

```
Loading required package: lavaan

This is lavaan 0.6-19
lavaan is FREE software! Please report any bugs.


Attaching package: 'lavaan'

The following object is masked from 'package:psych':

    cor2cov

Warning: lavaan->lav_data_full():
   Level-1 variable "y2" has no variance within some clusters . The cluster
   ids with zero within variance are: 409, 431, 501.

Warning: lavaan->lav_data_full():
   Level-1 variable "y3" has no variance within some clusters . The cluster
   ids with zero within variance are: 351, 451, 463.

Warning: lavaan->lav_data_full():
   Level-1 variable "y4" has no variance within some clusters . The cluster
   ids with zero within variance are: 351, 423, 447, 451, 471.

Warning: lavaan->lav_data_full():
   Level-1 variable "y2" has no variance within some clusters . The cluster
   ids with zero within variance are: 409, 431, 501.

Warning: lavaan->lav_data_full():
   Level-1 variable "y3" has no variance within some clusters . The cluster
   ids with zero within variance are: 351, 451, 463.

Warning: lavaan->lav_data_full():
   Level-1 variable "y4" has no variance within some clusters . The cluster
   ids with zero within variance are: 351, 423, 447, 451, 471.
```

```
Parallel analysis suggests that the number of factors =  NA  and the number of components =
Parallel analysis suggests that the number of factors =  NA  and the number of components =


$alpha
  alpha2l    alphab    alphaw
0.8342556 0.8218115 0.7679396

$alpha_ci
             2.5%     97.5%
alpha2l 0.7655752 0.8834255
alphab  0.7013840 0.8877646
alphaw  0.7117038 0.8122969

$omega
  omega2l    omegab    omegaw
0.8326827 0.8158880 0.7717208

$omega_ci
             2.5%     97.5%
omega2l 0.7566153 0.8828240
omegab  0.6723170 0.8870760
omegaw  0.7150554 0.8116054

$ncomp
 within between
      1       1
```

The values are labelled “alpha2l”, “alphab”, and “alphaw”, followed by the corresponding 95% CIs.