

(1) Krippendorff の α 係数の信頼区間を見積もる
(2) 一致度の低いコードがないかチェックする

v.2023-12-16 早大政経・中村理

このパッケージは R を用いて Krippendorff の α 係数の値を求めます。また、Bootstrap 法を用いて α 係数の信頼区間を求めます。さらに、どのコードが平均的に他のコードと一致度が高かったか低かったか、といった情報も提供します。

これまで、Krippendorff の α 係数の信頼区間を求めるには、有料の SPSS で、さらに K-Alpha というスクリプトを使う必要がありました¹。このパッケージは有料の SPSS にかえてフリーで利用できる R で、K-Alpha 以上のことをできるようにしました。このパッケージは、R を使ったことがある人、または R を使える人がそばにいる環境、であれば簡単に利用できるかと思います。

内容

1	はじめに	2
2	準備 1：インストール関連（2 回目以降は不要）	3
3	準備 2：コーディング・データファイル	4
4	実行	6
4.1	1 つ目の実行法	7
4.2	2 つ目の実行法（Windows のみ）	9
4.3	出力される結果	9
5	学び用	11
6	トラブル・シューティング	12

¹ SPSS を用いて Krippendorff の α を計算するには K-Alpha というスクリプトが必要です。なお、SPSS にはもともと信頼性の指標として「 α 係数」が計算できるよう組み込まれています。が、これは Cronbach（クロンバック）の α を求めるものです（2021-06 現在）。これはたとえば心理テストで被験者が類似の質問群に一貫性を持って答えたかを測る指標で、Krippendorff の α （Krippendorff 1970）とは概念も計算法も異なります。

Cronbach の α ：2 値変数に適用できる Kuder-Richardson（1937）20（KR20）を一般変数に拡張したもの（Cronbach 1951）。折半法だと質問群を 2 分割する方法により相関値が変わりますが、その全分割パターンのそれぞれで計算される相関値を平均することで係数値を算出します。

1 はじめに

- 注意点

- 以下、原則として1つもとばすことなく順に読み、必要なものを実行してください。
- 以下で、たとえば「> library(irr)」と「>」で始まるものは、「>」の後ろの部分をもその通りにRに入力し（コピーでOK²）、最後に enter キーを押してください（この例だと「library(irr)」）。そうすると、Rはそのコマンドが入力されたとして実行してくれます。

Rstudio を使っている場合は、「コンソール」と呼ばれるところへ入力してください。デフォルトのままであればおそらく左下にあるものがコンソールです。

- Windows の場合、フォルダの表示オプションで、ファイルの拡張子を表示させるようにしておくことを勧めます。そのやり方は「フォルダ 表示オプション 拡張子」で検索して自分で調べられます。調べても分からない場合はそのままでも可ですが、後々、混乱の原因になる場合があります。

- 当パックの同梱物

zip ファイルを展開すると、以下が含まれたフォルダが作られます：

➤ alpha.R	R スクリプト
➤ alpha_bat.bat	ドラッグ&ドロップ用の Windows バッチファイル
➤ data_test_KR_2coders.csv	テスト用コーディング・データファイル
➤ data_test_KR_2coders_factor.csv	テスト用コーディング・データファイル
➤ data_test_KR_3coders.csv	テスト用コーディング・データファイル
➤ data_test_KR_4coders.csv	テスト用コーディング・データファイル
➤ data_test_KR_4coders_factor.csv	テスト用コーディング・データファイル

² コピーでうまくいかない場合は、貼り付けた際に半角スペースが全角に変わってないか、半角コンマが全角に変わってないか、半角括弧が全角に変わってないか、改行部にスペースが入ってないか、チェックしてください。

- data_test_KR_4coders_factor_JP.csv テスト用コーディング・データファイル
- data_test_KR_5coders.csv テスト用コーディング・データファイル

- 計算について

- Bootstrap の R での実施法は以下を参照しました：
太郎丸 博「ブートストラップ入門」(2016 年 4 月 3 日)
<http://tarohmaru.web.fc2.com/R/bootstrapping.html> (ローカル保存済)
- Krippendorff の α 係数の計算には irr パッケージの `kripp.alpha()` を使っています。

2 準備 1: インストール関連 (2 回目以降は不要)

- PC の用意

- R をインストールできるものなら何でも結構です。

- PC に R をインストール

- Rstudio の使用経験があれば、Rstudio でも OK。
今日はじめて R を使うという方で、かつ隣に Rstudio の経験者がいない場合は、Rstudio でなく、R を勧めます。

- R を起動

R または Rstudio を起動します。
(以降、「R または Rstudio」は、R とのみ表現します)

- R 上で irr をインストール

R において以前に irr パッケージをインストールしていなければ、インストールします。

- R 上で以下を実行します。以前に済ませていれば今回する必要はありません。
> `install.packages("irr")`

このとき、「ミラーサイトを選べ」などと出てきたら、日本のもの、あるいは近いもの、を適当に選びます。

- Windows で R は動くがこのインストールはできない場合
R をいったん終了し、あらためて R を右クリックして「管理者権限で実行」してみてください。そのうえでもういちどこの `install.packages` を試みてください。
- **【重要】** 一度インストールすれば、今後 R を起動する際も二度とこれらの操作をする必要はありません。

3 準備 2：コーディング・データファイル

- **データファイルを csv で準備**

入力するデータを Excel など編集し、csv 形式で保存します。

その際、その csv ファイルのデータフォーマットは以下にしたがいます：

- 縦にコードが並び、
- 横に記録単位が並ぶ。
- その逆では絶対にない。
- 1 行目や先頭列にヘッダ行はつけない。データ値のみが入ったファイルにする。
- コーディングされなかった部分（データ欠損）は空欄にする。
逆に、コーディングした結果として空欄の場合は何らかの値で埋める必要有。
- データ値は数値型・因子型のいずれでも可。
数値型：コーディングの結果、たとえば 0, 1, 2 などの数値が記録されたもの。
因子型：コーディングの結果、たとえば「P」「N」などの文字列が記録されたもの。

4 人のコード（1～4 行目）が 12 の分析対象（A～L）をコーディングした結果の具体例は以下です（Krippendorff 2011 より）。空欄はコードがコーディングをしなかった部分です：

	A	B	C	D	E	F	G	数値バー	I	J	K	L	M
1	1	2	3	3	2	1	4	1	2				
2	1	2	3	3	2	2	4	1	2	5		3	
3		3	3	3	2	3	4	2	2	5	1		
4	1	2	3	3	2	4	4	1	2	5	1		
5													
6													

- **データファイルの例**

参考例として、以下のデータファイルをパックに含めています。

- data_test_KR_2coders.csv
クリッペンドルフ「メッセージ分析の技法」第1版12刷 p.209 の例
nominal の場合、 $\alpha=0.0952$
(値はテキストと同じ。参考まで、このデータは 0/1 の 2 値のため、ordinal、interval でも α は同じになる。)
- data_test_KR_2coders_factor.csv
上の 2coders と同じだが、数値型ではなく因子型でデータを収めたもの。
 α も上の 2coders と同じ³。
- data_test_KR_3coders.csv
クリッペンドルフ「メッセージ分析の技法」第1版12刷 p.219 の例
nominal の場合、 $\alpha=0.635$
(なぜか教科書の 0.6417323 と微妙に異なる。教科書の計算プロセスは正しいと中村が確認済。SPSS with Kalpha でも 0.6417 と出てくる。従って、kripp.alpha にバグがあるものと思われる。ちなみに R で interval だと 0.538)
- data_test_KR_4coders.csv
Computing Krippendorff's Alpha-Reliability
(by K. Krippendorff, 1-25-2011, U-Penn. Scholarly Commons より)
nominal の場合、 $\alpha=0.743$
interval の場合、 $\alpha=0.849$
(参照した資料通り。コーダのデータ中に Not Available を含む例)
- data_test_KR_4coders_factor.csv
上の 4coders と同じだが、数値型ではなく因子型でデータを収めたもの。
nominal で $\alpha=0.743$
(因子型のため interval/ordinal/ratio は指定不可。
これらを指定したければ自分であらかじめ数値に置き換えておくこと)
- data_test_KR_4coders_factor_JP.csv
上の 4coders と同じだが、数値型ではなく因子型でデータを収めたもの。
また、その因子を日本語（文字コードは Shift-JIS=ANSI=cp932）で収めたもの。読み込む際に文字コードの指定が必要になる。

³ 変数が nominal の場合、当スクリプトはデータをいったん数値に置き換えてから matrix() に変換している。理由は、読み込むファイルに記録されたデータが因子型 (0, 1 といったものでなく P, N といったもの) の場合、そのまま matrix() に変換すると、数値にはなるものの、期待するようには変換されないからである。たとえば data_test_KR_2coders_factor.csv の場合、このファイルから読み取った値を数値へ置き換えずに matrix() に変換して kripp.alpha() へ入力すると、nominal/ordinal で $\alpha=-0.1875$ と誤った値が出力される（正解は $\alpha=0.0952$ である）。

- data_test_KR_5coders.csv
4coders のデータに 5 人目のコードを追加したもの。ただし、5 人目はでたらめなコーディングをしている。
nominal の場合、 alpha=0.292

特に data_test_KR_4coders.csv に注目しておいてください。

これは、4 人のコードが 12 の文書をコーディングしたものの、一部のデータが欠損している（コーディングがされていない）という例です。

- **変数の型を確認**

あなたのコーディング変数の型は以下のどれか、確認します。

これはコーディング実施前に研究者がすでに決めているはずのものです。

- 名義尺度／順序尺度／等間隔尺度／比率尺度

- **変数の型を指定**

上で確認した変数の型に応じて、alpha.R 先頭部の v_type を編集します。

- デフォルトでは v_type は nominal に設定されています。このままでよければ編集は不要です。
- alpha.R は PC 上でファイルを右クリックするなどして編集できます。
Rstudio 上でも編集できます。
- 編集内容は以下の通り。

指定の詳細は R の irr パッケージの kripp.alpha を参照：

◇ 名義尺度	nominal
◇ 順序尺度	ordinal
◇ 等間隔尺度	interval
◇ 比率尺度	ratio

（よく分からなければひとまず nominal にしておくのが良いかと思います）

- **【注意】** alpha.R を編集した際は保存することを忘れないでください。
保存することによって編集した内容が反映されます。
- **【注意】** エラーを防ぐため、alpha.R の他の部分は編集しないことを勧めます。

4 実行

2 通りの実行法があります。2 つ目は Windows のみです。

4.1 1つ目の実行法

- **フォルダへ行く**

上ですでにパッケージの zip ファイルを展開し、必要に応じて alpha.R を編集しましたね。そのフォルダに入ります。

- **データファイルを置く**

そのフォルダにコーディング・データファイルを「data_test.csv」という名前で置いてください。

- まず動作確認だけしたい場合は、パッケージ内の「data_test_KR_4coders.csv」をコピーして「data_test.csv」という名のファイルを作り、お試しください。

- **R を起動**

alpha.R をダブルクリックし、R を起動します。

- 繰り返しになりますが、この章を通じて R は以下のいずれかを指します：

Rstudio/Rgui/R

- 「どのプログラムで alpha.R を開くか」と聞かれる場合は R を選択してください。
- R 以外のもので alpha.R が開かれる場合は、R を選択して開いてください。Windows の場合、右クリック→プログラムの選択→Rstudio、など。
- alpha.R をダブルクリックすることなく R を起動した場合は、R をいったん終了させ、alpha.R をあらためてダブルクリック⁴してください。

- **R の動作する場所をデータを置いたフォルダに設定する**

ここは鬼門です..

R の作業ディレクトリを確認します。

作業ディレクトリとは、R がファイルの入出力をするフォルダのようなものです。

- R 上で以下を実行してください。

```
> getwd()
```

⁴ R をいったん終了させ、alpha.R をダブルクリックして起動し直す理由は、作業ディレクトリを現在のディレクトリにするためです。ですので、自分で作業ディレクトリを変更できる方は R を終了することなく作業ディレクトリを変更すれば OK です。

- この結果が、alpha.R を置いているフォルダと同じ場合、問題ありません。良かったですね。先へお進みください。
- 異なる場合、以下の手順で作業ディレクトリを alpha.R を置いているフォルダへ変更します。分からなければ、R に精通している人の手を借りてください。

✧ **Rstudio の場合：**

メニューから次のようにたどるなどして、設定します。

Session > Set Working Directory > Choose Directory

→フォルダを指定

✧ **Rgui の場合：**

メニューから次のようにたどるなどして、設定します。

ファイル > ディレクトリの変更

→フォルダを指定

✧ **R の場合：**

R 上で以下を実行して設定します。

> setwd("<フォルダの場所>")

Windows の例（実際のフォルダの場所にしてください）：

> setwd("C:/Users/myname/Downloads/00pack_v20230111")

> setwd("C:/Users/myname/Desktop/00pack_v20230111")

Mac の例？：

> setwd("/Users/myname/00pack_v20230111")

● **変数の型の確認**

コーディング変数の型に応じて、alpha.R 内の v_type を編集します。上で済ませている場合はとばしてください。手順の確認のため、ここにも入れました。

- alpha.R を開いて文字化けしていても、v_type のみ編集すれば OK です。

● **alpha.R の実行**

R の上で以下をタイプし、Enter キーを押します：

> source("alpha.R")

※このとき、エラーが出る場合は、めげずにエラーメッセージを読んでください。そして、あるはずの data_test.csv がないと言われる場合、以下を確認してください：

- ✓ 上でおこなった getwd() の結果が正しかったか。
- ✓ 拡張子を間違えてないか。フォルダの設定で拡張子を非表示にしている

ことに気づかず変なファイル名になっていないか。

※データファイルが日本語で文字化けする場合

- ✓ 最も簡単な対処法は、データファイル中の日本語を英数字に置き換えることです。
- ✓ 次に簡単な対処法は、alpha.R 内の `read.csv()` に、`fileEncoding="cp932"` (や "UTF-8") などとデータファイルの文字コードを指定することです。
- ✓ 次に簡単な対処法は、データファイルの文字コードを、あなたの R と同じにすることです。UTF-8、cp932などを試してください。

- **計算結果を得る**

結果が R 上に出力されます。結果の見方は 4.3 を参照。

この方法のメリット： シンプル。多分エラーは出ない。
この方法のデメリット： データファイル名が `data_test.csv` に固定される。
わずかだがコマンドラインでの作業になる。

4.2 2つ目の実行法 (Windows のみ)

- パッケージの zip ファイルを展開し、できたフォルダに入る。
- そこにコーディング・データファイルを置く。名前はなんでもよい。
- コーディング変数の型に応じて、alpha.R 内の `v_type` を編集する (上を参照)。
 - alpha.R を開いて文字化けしていても、`v_type` のみ編集すれば OK。
- コーディング・データファイルを `alpha.bat.bat` へドラッグ&ドロップする。
- 結果が出力される。結果の見方は 4.3 を参照。

この方法のメリット： ドラッグ&ドロップだけで処理できる。
データファイル名はなんでもよい。
この方法のデメリット： エラーに会うかもしれない。

4.3 出力される結果

- **結果の読み方**

出力される結果は以下です：

1. Krippendorff's alpha

Krippendorff の α 係数の値です。

2. Bootstrap distribution

ブートストラップ法による、下から 0.5%目、2.5%目、5%目、50%目、95%目、97.5%目、99.5%目の α 係数の値。信頼区間の推定に用います。

- ◇ 例えば 95%信頼区間を知りたい場合、2.5%目から 97.5%目をとれば OK。
例：以下の表示だと、0.5550--0.7633 が 95%信頼区間

```
Bootstrap distribution:(0.005 0.025 0.05 0.5 0.95 0.975 0.995)
0.5151 0.5550 0.5703 0.6597 0.7468 0.7633 0.7957
```

- ◇ ブートストラップ法はランダムサンプリングを使うので、実行するたびに結果が変わります。その変動具合をより小さくしたければ、alpha.R 内の n_boot を大きな値にしましょう。

3. Inter-coder test

コーダ間の一致度の状況

- ◇ たとえば以下の表示は、001 人目のコーダと 002 人目のコーダの間のみで α 係数を計算すると 0.852 になることを示します。また、2 人の単純な一致度⁵は 0.667 になることを示しています。

```
Coder 001 vs. Coder 002 Kripp-alpha=0.852 Simple-agreement= 0.667
```

- ◇ たとえば以下の表示は、「002 人目のコーダでは、他のコーダとペアを組んだときの α 係数の平均値が 0.73 になる。それは、他のコーダでの同様の値と比べ、3 番目に良い値である。」ことを示します。

```
Coder 002 rank= 3 average=0.73
```

- ◇ 仮に Coder が計 5 人いて、あるコーダの順位が rank=5 である場合、そのコーダは、5 人の中でもっとも他者との一致状況が低かったことを意味します。一般に、そうしたコーダを除外する必要はありません。しかし、特

⁵ ここでの単純な一致度とは、(一致した分析対象数) ÷ (全分析対象数) で算出しています。「一致した分析対象数」は欠損値 (NA) を除外しています。「全分析対象数」は欠損値であった分析対象も含みます。

定のコードが他のコードのすべてと著しく低い成績である場合は、そうしたコードの扱いを検討したほうがよいかもしれません。

data_test_KR_5coders.csv はそうした例です。このファイルは data_test_KR_4coders.csv に 5 人目のデータを追加したものです。この 5 人目はデタラメなコーディングをしており、 α 係数は他の誰とでも著しく低い値になっています。

- **論文に信頼区間を記すとき**

- たとえば、「Krippendorff の α 係数の値は 0.68、Bootstrap 法による 95% 信頼区間は $0.52 \leq \alpha \leq 0.84$ であった。」などと記すことになります。

- **Bootstrap による α 係数の分布図**

- R のコンソール上でなら、source("alpha.R")を実行後に hist(a_boot)でみることができます。

5 学び用

- **計算結果の確認**

ゼミ演習で初めて使う場合、以下を順に試し、結果が一致するか、チーム内で互いに確かめてください。

- 2coders のデータ、変数が名義尺度のとき
- 3coders のデータ、変数が名義尺度のとき
- 4coders のデータ、変数が名義尺度のとき／変数が順序尺度のとき

- **変数の尺度 1**

以下をチームごとに調べて確認してください。

- 名義尺度とはなにか
- 間隔尺度とはなにか
- 順序尺度とはなにか

- **変数の尺度 2**

- 以下の変数の尺度は何でしょうか。名義尺度、間隔尺度、順序尺度の中から答えてください。チーム内で見解は一致するでしょうか。
- 登場人物の「年齢」という変数。実年齢を数値で記録。
- 登場人物の「年齢」という変数。

カテゴリを「10代／20～30代／40～50代／60以上」に分けて記録。

- 登場人物の「職業」という変数。
カテゴリを「専門家／政治家／行政／その他」に分けて記録。
- 登場人物の「職業」という変数。
カテゴリを「専門家=1／政治家=2／行政=3／その他=4」に分けて数値を記録。
- 記事の「論調」という変数。
カテゴリを「肯定的／両論併記／否定的／言及なし」に分けて記録。
- 記事の「論調」という変数。
カテゴリを「肯定的／中立／否定的」に分けて記録。

- **コードが2名のときの Krippendorff の α と他の指標との関係**

Hayes and Krippendorff (2007)⁶p. 82 にある通り、2 コード間で Krippendorff の α を求める場合は、他の指標と以下のように関係することになります：

- 名義尺度の場合、漸近的に Scott の π 経数に近づく。
- 順位尺度の場合、Spearman の順位相関係数 ρ と同じ。

6 トラブル・シューティング

- うまくいかない場合、何のエラーが表示されているか、まず読み取ってください。
- 1 目の実行法で、source を実行すると alpha.R や data_test.csv が見つからないとエラーが出る場合
 - R 上で getwd() とタイプする。
 - するとあなたが今 R を動かしているフォルダが表示される。
 - それがもともと alpha.R と data_test.csv を置いているフォルダと異なる場合は、alpha.R と data_test.csv を置いているフォルダに R の作業ディレクトリを移す。
(または、getwd() ででてきたフォルダに alpha.R や data_test.csv を移す)
- 2 目の実行法で、Windows で、alpha.bat.bat にドラッグ&ドロップしても

⁶ Hayes and Krippendorff, *Communication Methods and Measures*, 1(1)(2007), pp. 77–89

Rscript.exe が見つからないというエラーが出る場合

R 上で source を使う（上の 1 つ目の実行法）ことを推奨する。それでもドラッグ & ドロップを使いたければ、以下の手順で Rscript.exe までの絶対パスを alpha_bat.bat に決め打ちする（難しいので推奨しない）。

- R がどこにインストールされているか、確認する。
 - ✧ 例：C:\Program Files\R\R-4.0.3
- そのフォルダ内に bin というサブフォルダがあり、その中に Rscript.exe というファイルがあることを確認する。
- 上で確認した Rscript.exe までのパスを、当バック内の alpha_bat.bat に書き込む。
 - ✧ alpha_bat.bat は、右クリックして「編集」を選ぶと編集できる。
 - ✧ alpha_bat.bat 内に「Rscript.exe までのパスを決め打ちする場合は以下」と記してある。その続きに以下のように書く。また、そこまでのラインはすべて消す。
 - ✧ 記入例（あなたが編集するのは下線部だけだと思う）：
"C:\Program Files\R\R-4.0.3\bin\Rscript.exe" "%~dp0/alpha.R" %1