

13日目：尺度作成（ α 係数など）

さて、いろいろと分析を試し、結果を比較してみていただけただけでしょうか？ 今回のサンプルデータは、まあまあやりやすい部類ではないでしょうか。固有値の推移や平行分析による示唆などから、因子数3が最適だろうと思います。4すると、かなり小さなまとまりを抽出することになりますし。抽出方法を最尤法、回転をプロマックスで指定した結果を基準に、方法を変えながら結果を比較しました。いろいろやってみると、固まってあまり動きがない項目や、その周辺でうろろする項目などがわかると思います。たぶん…

今回はRの使い方の資料なので、どれを採用するのが良いのかという話は少し横に置いておきます。私の好みからすると最尤法、独立クラスター回転 ("cluster") の結果かと思えますので、これを使って、本日は尺度作成に向かう作業を説明します。

因子分析の結果をエクセルで整理すると、右のようになります。

因子パターンが.400以上を緑系の網掛け、.300以上をオレンジ系の網掛けにしてあります。.400以上を2つの因子に対して示す項目はありませんが、どの因子にも.400以上を示さない項目はありません。

もう少し単純構造に近づきたいので、いずれにも高いパターンを示さない項目、複数に高いパターンを示す項目を中心に削除をしたうえで因子分析を繰り返す作業を行いました。

item	ML3	ML1	ML2
b12 有名な	.730	.137	-.023
b17 安定している	.707	.139	.046
b19 便利な	.664	-.315	-.088
b9 使いやすいそう	.646	-.210	-.065
b7 こだわりがある	.638	.220	-.056
b4 操作性のよい	.594	.230	.037
b15 安心感のある	.577	.044	.349
b2 おしゃれな	.410	.334	-.204
b16 そそられる	.346	.340	.024
b20 古典的な	.185	-.775	-.191
b14 都会的な	.078	.752	.080
b5 洗練された	.069	.732	.183
b13 機能的な	-.057	.608	-.062
b1 親近感のある	-.029	.530	-.131
b10 かわいい	.288	.290	-.116
b11 近寄り難い	-.012	-.159	.895
b6 高価な	.044	-.005	.782
b3 高級感のある	.003	-.145	.744
b18 りっぱな	.002	.132	.494
b8 無機質な	-.041	.341	.359

最終的に、次の15項目を使い、最尤法、独立クラスター回転を行った結果を採用します(次表参照)。オレンジ系の網掛けがなくなり、かなり単純構造に近い結果を得られたと思います。

それぞれの因子ですが、1番目の因子は、帰納や操作に関する項目が高いパターンを示しています。そこで「機能性」と命名しておきます。

2番目の因子は「古典的な」が負のパターンを示し、「都会的な」などが正のパターンを示

しています。そこでこれを「デザイン性」と命名しておきます。

3番目は「近寄りやすい」「高価な」「高級感のある」などが高いパターンを示しています。そこでこれを「グレード感」と命名しておきます。

ダミーのデータなので、命名はちょっと適当ですが…

	item	ML1	ML3	ML2
b12	有名な	.843	-.022	-.067
b17	安定している	.842	-.014	-.014
b7	こだわりがある	.757	.087	-.117
b15	安心感のある	.705	-.070	.287
b4	操作性のよい	.661	.139	-.004
b9	使いやすそう	.546	-.185	-.062
b20	古典的な	.139	-.804	-.125
b14	都会的な	.095	.766	.021
b5	洗練された	.103	.720	.135
b13	機能的な	-.061	.641	-.087
b1	親近感のある	-.064	.563	-.132
b11	近寄り難い	-.041	-.080	.906
b3	高級感のある	-.058	-.046	.760
b6	高価な	.084	.028	.735
b18	りっぱな	.036	.155	.430

この因子分析結果をもとに、各因子を代表する項目を選出します。今回は、ひとつの因子に .400 以上を示し、他の因子に .300 未満のパターンを示す項目を選出することにします。これで下位尺度を構成する候補とします。

ここまで来ると、後の手順は大体想像できると思います。α係数を出し、それが満足できる値なら合計点を算出し、基礎統計量を求めてみる…。こういう流れが論文では一般的ですね。

まずはα係数からいきます…とりたいところなのですが、因子分析結果をみるとわかるように、1つ逆転項目 (b20) が入っています。まずはこれを逆転させます。

命令はとても単純で、代入を使います。4件法で、1から4点で得点化されているので、それを5から引けば逆転できます。

```
x$br20 <- 5-x$b20
```

これでファイル x に br20 という項目が追加され、そこに x の b20 のデータを 5 から引いたものが入ります。br20 という新しい項目名を作らなくとも、ここを b20 <- 5-x\$b20 とすればデータの上書きをしてくれるのですが、後で逆転済みだったかどうか混乱するとどうしようもないので、新しい変数を作っておくことをおすすめします。

うまく変換できているかどうかを確認する方法はいくらでもあると思いますが、table(x\$br20, x\$b20) というのをやってみるのも面白いと思います。

うまくいっていれば、右図のように、右上がりの対角線上にのみデータが位置するはずです。

これで逆転項目の処理も終わりましたので、いよいよ

```
> table(x$br20, x$b20)
```

```

      1  2  3  4
1  0  0  0 63
2  0  0 72  0
3  0 55  0  0
4 108  0  0  0

```

α 係数です…が、もう一つその前に。

これからの分析を考えると、因子分析の前にやったように、各下位尺度の候補をまとめてひとつのまとまりにしておくとか何かと便利です。そこで、以下のような感じで3尺度ごとのファイルを作成しておきます。

```
label_f1 <- c("b12", "b17", "b7", "b15", "b4", "b9")
```

```
label_f2 <- c("br20", "b14", "b5", "b13", "b1")
```

```
label_f3 <- c("b11", "b3", "b6", "b18")
```

```
dat_f1 <- x[label_f1]
```

```
dat_f2 <- x[label_f2]
```

```
dat_f3 <- x[label_f3]
```

さて、いよいよ α 係数です。これは **psych** パッケージに入っています。

```
alpha(dat_f1)
```

これで **dat_f1**、つまり「信用・信頼」因子を構成する項目群の α 係数を算出してくれます。各種の指標も同時に出力してくれるので、とても簡単です。

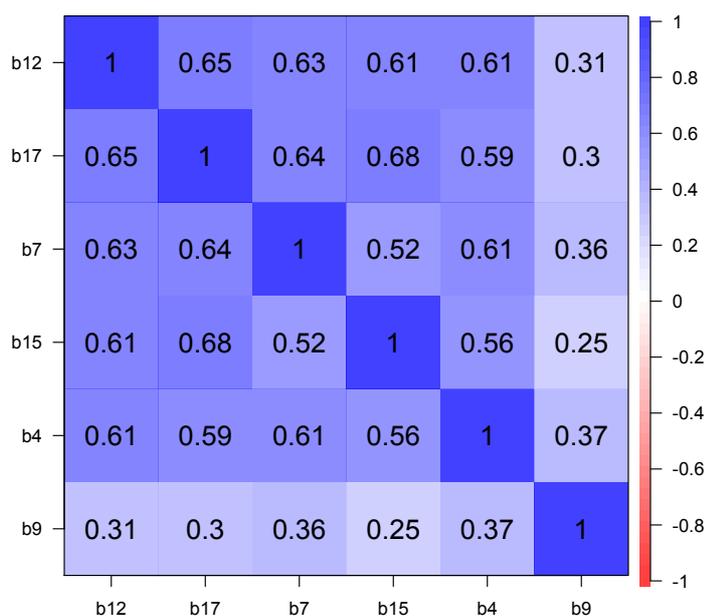
なお同時に項目間の相関係数を出しておくともよいでしょう。

```
cor(dat_f1, use="complete.obs")
```

またこれも **psych** パッケージに入っていますが、項目間の相関係数を使って、**cor.plot** でその関係を図示しておいても項目の検討がやりやすくなると思います。

```
cor.plot(cor(dat_f1, use="complete.obs"), numbers=TRUE)
```

変数間の相関係数の値で色をコントロールして、右のような図を作ってくれます。**numbers=TRUE** を入れることで、相関係数を表示させています。**cor()**の結果を読み取るのはどうも苦手、という人には良いかと思います。また、これを併用すると誤認が減りそうな気がします。



さて、`alpha` で出力される指標がかなり多いです。うれしい悲鳴ですが…

それぞれの詳細はヘルプで確認してください。

<code>raw_alpha</code>	α 係数 (論文への記載はこれ)
<code>std.alpha</code>	標準化された α 係数
<code>average_r</code>	項目間相関の平均値

その下の一覧が、**Reliability if an item is dropped**。つまり、その項目を削除した時の各指標です。主として、ここの指標との比較から考えていけばよいでしょう。

さらに下にあるのが、**Item statistics**。訳すまでもないでしょう。この一覧、今一つ内容がわからないものも含まれますが、注目しておく代表的な指標としては以下のものでしょう。

<code>r.drop</code>	その項目を除いた場合の合計と、その項目の相関
---------------------	------------------------

さて、この α 係数の結果の中で、最初の、`raw_alpha` や `std.alpha` は、小数点以下1桁で表示される場合もあります。今回だと、3つ目の因子は以下のよう。

```
Reliability analysis
Call: alpha(x = dat_f3)

raw_alpha std.alpha G6(smc) average_r S/N ase mean sd median_r
0.8        0.8        0.77        0.49 3.9 0.018 2 0.8 0.47
```

実際は小数点以下3桁目で丸められているようなのだが、ちょっと気持ち悪い。もう少し桁数をみたいなら、2日目に使った `print` を使うのが簡単です。

```
print(alpha(dat_f3), digits=5)
```

こうやっておくと、以下のような詳しい数値が出てきます。

```
Reliability analysis
Call: alpha(x = dat_f3)

raw_alpha std.alpha G6(smc) average_r S/N ase mean sd median_r
0.80041 0.79637 0.77127 0.49437 3.9109 0.017872 2.0201 0.80115 0.4692
```

さて、このように簡単に α 係数を算出してくれるのですが、ちょっと気を付けておかなければならないところをひとつ。特に少し前のバージョンを使っている場合です。

ためしに、逆転項目である **b20** を逆転しないまま使って、2 番目の因子「デザイン性」の α 係数を算出してみてください。

`raw_alpha` が **0.18** となっていればよいのですが、以前にはこれが **0.82** になるバージョンもありました。逆転項目のようであれば、自動的に逆転させて計算してくれたのです。`check.keys=`というオプションの設定によるものですが、最近のバージョンではデフォルトが **FALSE** になっています。これを **TRUE** にすると、逆転項目を自動的に逆転させて計算します。自動的に逆転した場合、**Reliability if an item is dropped** のリストで、**b20** の部分に「-」とつきます。また結果の最後に、「いくつかの項目で合計と負の相関が出たので自動的に逆転しました」というような警告が英語で出ます。これを見逃してしまうと、とても残念な誤解になってしまうので注意してください。

以下を使って、どのような結果になるか確認しておいてください。

```
label_f2r <- c("b20", "b14", "b5", "b13", "b1")
dat_f2r <- x[label_f2r]
alpha(dat_f2r, check.keys=TRUE)
```

最近では、 α 係数に加えて ω 係数も報告されている論文が増えています。`psych` パッケージで計算できますので紹介しておきます。

```
omega(dat_f1)
```

以上です。あっけないほど簡単です。

さて、これらの結果を眺めてみると、「機能性」の α 係数は.86、「デザイン性」が.82、「グレード感」が.80 という値です。 α 係数自体も満足のいく値なので、これで項目を確定します。

これで各下位尺度の合計点を出せばよいのですが、ここもいろいろなやり方ができます。きわめて無難な線なら…

```
x$total_f1 <- x$b12 + x$b17 + x$b7 + x$b15 + x$b4 + x$b9
```

などと、新しい変数名をつくり、素直に計算式を書くというやり方でしょうか。

もう少し楽にとということなら、一見、ちょっとわかりにくいかもしれませんが、結構簡単なのは以下ではないかと…

```
x$total_f1 <- rowSums(x[, label_f1])
x$total_f2 <- rowSums(x[, label_f2])
x$total_f3 <- rowSums(x[, label_f3])
```

`rowSums(x[,label_f1])`は、その名の通り、行の合計（1ケースのデータは1行に
ているため行の合計です。列と勘違いしないように。）を求める命令で、`x[,label_f1]`
と、`x`の中の`label_f1`で指定された列のみを使ってそれを計算しなさいということにな
ります。つまり各ケースの合計が`total_f1`にストックされます。同様に`label_f2` ,
`label_f3`についても合計を計算します。

少しイメージがわきにくいかもしれませんが、`x`の中身（変数名など）を見てみると、
`total_f1`などの新しい変数が列の最後に加えられていると思います。なお、`rowSums`は
欠損値がある場合は欠損値を返すので、`b1`に欠損値を持つ場合の`total_f2`もNAになっ
ています。

なお、合計を項目数で割ったものを合計得点にしたければ…

```
x$total_f1 <- rowSums(x[,label_f1]) / 6
```

などとしておけばよいです。

本日はここまでです。