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でしょうか？　今回のサンプルデータは，結構やっかいな部類（きれいな構造ではないと言うべきか…）に入ると思います。たとえば，因子数３，プロマックス回転を採用し，抽出方法のみを変更して，主因子法，最尤法，一般化最小二乗法の結果を比べてみると，かなり結果が違ってきます。

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の結果かと思いますので，これを使って，本日は尺度項目を確定し，内部一貫性を求め，合計得点を計算するとい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。

		ML3	ML1	ML2
b12	有名な	.741	.109	−.005
b17	安定している	.721	.121	.059
b7	こだわりがある	.652	.191	−.048
b19	便利な	.648	−.350	−.019
b9	使いやすそう	.635	−.241	−.010
b4	操作性のよい	.612	.215	.035
b15	安心感のある	.596	.072	.347
b2	おしゃれな	.423	.292	−.209
b16	そそられる	.369	.331	.000
b10	かわいい	.301	.264	−.127
b20	古典的な	.136	−.808	−.087
b14	都会的な	.123	.761	−.003
b5	洗練された	.117	.755	.095
b13	機能的な	−.026	.602	−.126
b1	親近感のある	−.004	.513	−.181
b8	無機質な	−.008	.391	.297
b11	近寄り難い	.014	−.037	.853
b6	高価な	.074	.099	.733
b3	高級感のある	.023	−.045	.711
b18	りっぱな	.028	.199	.447

因子パターンが.400 以上を赤字にしてあります。赤字を２つの因子に対して示す項目はありませんが，どの因子にも赤字を示さない項目はあります。

それぞれの因子に関しては，１番目の因子は，利用・活用に関する項目が高いパターンを示しています。そこで「実用的」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「近寄りがたい」「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「高級感」と命名しておきます。

ダミーのデータなので，命名にこだわってもしょうがないのですが…

これらの因子が見出されたとし，また因子パターンが.400 以上を基準として項目を抽出し，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…。こういう流れですね。

まずはα係数からいきます…と言い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，かつ，これまでにも出てきたものと同じイメージです。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とすればデータの上書きをしてくれるのですが，後で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) というも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。

```
> 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
>
```

これで逆転項目の処理も終わりましたので，いよいよ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。

```
l.f1 <- c("b2","b4","b7","b9","b12","b15","b17","b19")
l.f2 <- c("b1","b5","b13","b14","br20")
l.f3 <- c("b3","b6","b11","b18")
d.f1 <- x[l.f1]
d.f2 <- x[l.f2]
d.f3 <- x[l.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(d.f1)
```

これでd.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。

ですが，出力される指標がかなり多いので，うれしい悲鳴ですが…

出力の上から順に説明すると

raw_alpha	α係数
std.alpha	標準化されたα係数
G6(smc)	ガットマンのラムダ6
average_r	項目間相関の平均値
mean	平均
sd	標準偏差

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あるのですが，代表的な指標としては以下のものでしょう。

r.drop	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。

```
raw_alpha std.alpha G6(smc) average_r mean  sd
      0.8       0.8    0.77       0.49    2 0.8
```

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，いったんα係数の結果を何かに代入しておく。そして，ここの情報には total という名前がついているのでそれをよび出してやる。

```
alpha(d.f3) -> dd.f3
dd.f3$total
```

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。

```
> alpha(d.f3) -> dd.f3
> dd.f3$total
 raw_alpha std.alpha   G6(smc) average_r      mean        sd
 0.8004078 0.7963728 0.7712656 0.4943708 2.020134 0.8011549
```

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。

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

```
 raw_alpha std.alpha G6(smc) average_r mean   sd
      0.82      0.82    0.81      0.48  2.6 0.49

 Reliability if an item is dropped:
     raw_alpha std.alpha G6(smc) average_r
b1        0.83      0.83    0.80      0.55
b5        0.75      0.76    0.73      0.44
b13       0.80      0.80    0.78      0.50
b14       0.75      0.76    0.73      0.44
b20-      0.78      0.78    0.76      0.47

 Item statistics
       n    r r.cor r.drop mean   sd
b1   297 0.65  0.52   0.45  2.6 0.97
b5   298 0.83  0.80   0.73  2.9 1.04
b13  298 0.73  0.62   0.56  2.5 0.90
b14  298 0.83  0.80   0.73  2.7 0.97
b20- 298 0.78  0.72   0.64  2.3 1.17
```

これが，驚くことに右のようにばっちりと逆転して計算してくれるのです。違うところといえば，b20 の部分に「-」とついているところです。これを見逃してしまうと，とても残念なことになってしまうので注意してください。

こういう場合，結果の最後に警告が出ます。

「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」って，これは自動的にやらない方がいいと思うのだけれど…

```
警告メッセージ：
In alpha(d.f2) :
  Some items were negatively correlated with total scale and were automatically reversed
```

さて，これらの結果を眺めてみると，「信用・信頼」のα係数は.86，「デザイン」が.82，「高級」が.80 という値です。項目別にみると，ちょっと（かなり，というべきか…）気になるものもありますが，練習ですし，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$total.f1 <- x$b2 + x$b4 + x$b7 + x$b9 + x$b12 + x$b15 + x$b17 + x$b19
```

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…

```
to.f1 <- rowSums(x[,l.f1])
to.f2 <- rowSums(x[,l.f2])
to.f3 <- rowSums(x[,l.f3])
xx <- data.frame(x, to.f1, to.f2, to.f3)
```

rowSums(x[,l.f1])は，その名の通り，行の合計（1ケースのデータは1行に入っ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[,l.f1]と，x の中の l.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 to.f1 にストックされます。同様に l.f2 ，l.f3 についても合計を計算しておいて，data.frame(x, to.f1, to.f2, to.f3)で x にくっつけてやり，それに xx という名前をつけるという作業になります。

少しイメージがわきにくいかもしれませんが，xx を見てみると，以下のようにちゃんと新しい変数が結合されています。また rowSums は欠損値がある場合は欠損値を返すので，b2 に欠損値を持つ場合の to.f2 も NA になっています。

```
> head(xx)
    ID. 性別 学年 専攻 b1 b2 b3 b4 b5 b6 b7 b8 b9 b10 b11 b12 b13 b14 b15 b16 b17 b18 b19
1 10003    2    3    1  4  4  2  4  3  3  4  2  3   4   1   4   4   3   3   4   3   1   4
2 10004    2    3    1  4  3  2  4  3  1  2  2  1   1   2   3   2   3   2   1   2   1   2
3 10010    2    3    1 NA  4  1  4  4  1  4  1  4   4   1   4   4   4   4   4   4   1   4
4 10018    1    2    1  2  4  1  4  3  1  2  1  3   3   2   4   3   3   3   4   3   1   3
5 10019    1    2    1  3  2  3  4  2  3  4  1  3   2   3   2   3   2   2   2   3   2   3
6 10020    2    3    1  4  4  1  4  3  1  4  2  3   4   2   4   3   3   3   4   3   1   3
  b20 br20 to.f1 to.f2 to.f3
1   1    4    29    18     7
2   2    3    19    15     6
3   1    4    32    NA     4
4   3    2    26    13     5
5   4    1    23    11    11
6   2    3    28    16     5
```

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…

```
to.f1 <- rowSums(x[,l.f1]) / 8
```

などとしておけばよいです。

本日はここまでです。