

心理調査における尺度構成のためのRスクリプト

R script for Scale Construction in psychological survey

緒賀 郷志 (岐阜大学教育学部) Satoshi OGA

はじめに

心理学研究において利用頻度が高まっている統計環境ソフトR (R core team, 2014) であるが、Rはスクリプトを書くことでもって実行するため、初心者にとってとりつきにくい点は否めない。緒賀 (2015) はそういう初心者のためのパス解析を実行するスクリプトと、それらのコマンドの解説をおこなった。引き続き、本稿では、初心者がRを利用して心理調査データを分析する際、とくに尺度構成に関わる因子分析と信頼性係数の算出のスクリプトを紹介する。

Rスクリプトと解説

まず前提として、data.csv のファイルに以下の55個の変数が入っていることとする。ID, daigaku, gakunen, sex, age, a1-a30, b1-b20, である。daigaku の変数は、A, B, C でコーディングされ、gakunenは1,2,3,4 でコーディングされているとする。sexの変数は、男性はM, 女性はFでコーディングされているとする。またそれぞれの変数に欠損値がないという前提で分析をすすめる。a1-a30は3因子を想定して作成した尺度項目、b1-b20は2因子からなる既存の尺度項目とする。また質問項目は7件法という仮定とする。実際には、4件法、5件法などが多く使われているが、リッカート尺度は本来7件法以上が望ましいとされている。なお因子分析をする際に、2件法の場合はここでは紹介しないが、カテゴリカル因子分析というのを行うと知っておいて欲しい。

CSVファイルから、R環境内のデータフレームへの読み込み方法については緒賀 (2015) 同様に、次のスクリプトを使う。説明については、緒賀 (2015) を参照して欲しい。

```
(wd <- getwd( ))
x <- read.csv(paste0(wd, "/data.csv"))
```

さて、この読み込んだデータフレーム x に、NA (欠損値) があるかのチェックはanyNA関数で確認が可能である。

```
anyNA(x)
```

もし TRUE と表示されたならば欠損値がどこかにある。その場合、もとのデータに戻って入力漏れがないかの確認や、もともと欠損した値がある場合は、最頻値あるいは平均値などを代入する。も

しくは、下記のように `na.omit` 関数を用いて、欠損データを削除し、欠損値が含まれていないデータフレームで分析を続けることを勧める。なお、初心者のためのもので躓く可能性を減らすために上記を勧めるが、各関数で欠損値の取り扱いの方法もそれぞれ指定できるということは知っていて良いだろう。

```
x <- na.omit(x)
```

データフレーム `x` に問題なくデータが読み込まれているかを、下記の二つの関数を用いて、最初の部分と最後の部分を確認すると良い。

```
head(x)
tail(x)
```

次に、データフレームに入っている行と列の数を調べる。すなわち分析する人数と変数の数を確認する。

```
dim(x)
```

さらに、変数名を確認する。

```
names(x)
```

回答者の属性について確認するスクリプトは以下のとおりである。
最初に下記の下準備を行う。

```
x$daigaku <- as.factor(x$daigaku)
x$gakunen <- as.factor(x$gakunen)
x$sex <- as.factor(x$sex)
```

通常は大学 `daigaku`、性 `sex` はすでにファクターとして読み込まれているが、上記では念の為に `as.factor` 関数を用いて、そのデータの `class` を変更している。学年 `gakunen` は、数字として読み込まれているので、上記の手続きは必須である。

```
table(x$daigaku)           # 大学ごとの調査人数の算出
table(x$gakunen)         # 学年ごとの調査人数の算出
with(x, table(daigaku,   # 大学と学年にもとづく人数のクロス表の算出
              gakunen))
```

```
table(x$sex) # 性別ごとの調査人数の算出
```

table のコマンドは、その名前の通り factor ごとの人数の表を返してくる関数である。ダラーマーク \$ は、変数を指定する際に、データが入っているデータフレームの名前を頭につけて、分析する変数を指定するのに用いる。すなわち x\$sex は、x というデータフレームの中にある sex という変数を指定している。それとは別に、with 関数を用いての変数の指定方法もある。with 関数の最初の引数にデータフレームのオブジェクト名を指定することで、続くカッコ内の関数で用いる変数名は、x\$ を付ける必要がなくなる。複数の変数名を指定する時には、with 関数を用いたほうが便利であろう。

```
with(x, table(daigaku, sex)) # 大学と性別にもとづく人数のクロス表の算出
with(x, table(gakunen, sex)) # 学年と性別にもとづく人数のクロス表の算出
with(x, table(daigaku, gakunen, sex)) # 大学と学年と性別にもとづく人数のクロス表の算出
```

年齢の基本統計量を求める際には、追加パッケージである psych パッケージ (Revelle, 2015) 内にある関数を用いるのが便利である。また後述の因子分析にも psych パッケージ内の関数を紹介する。

```
library(psych) # psychパッケージの呼び出し
describe(x[5])
```

```
# xのデータフレームの5列目, すなわちage変数を指定している。あるいは下記でも良い。
describe(x$age)
with(x, describe(age))
```

```
describeBy(x$age, x$sex) # 性別ごとの年齢の基本統計量の算出
```

以上の describe 関数, describeBy 関数が基本統計量を算出する関数である。以上が、調査協力者の属性についての情報を得るスクリプトとなる。

次に作成した質問項目の回答のチェックを実行するスクリプトである。最初に分析する変数のみを取り出したデータフレームを作成し、そのオブジェクトを分析すると手際が良くなる。3 因子を想定して作成した尺度項目 a1 から a30 までの変数をとりだすことを考えると、これらは x の 6 列目から 35 列目までにしまわれている。そこで、6 から 35 というのを 6:35 というコロンで挟んだ形で表現し、あらたに x.a というオブジェクト (データフレーム) を作成する。作成されたのを確認するのに head 関数を用いてみよう。

```
x.a <- x[6:35]
head(x.a)
```

回答に偏りがないかを調べるために、箱ひげ図を描いてみる。箱ひげ図を描く関数は `boxplot` 関数である。

```
boxplot(x.a)
```

さらにわかりやすくヒストグラムで回答分布を描いてみる。検討中の項目数は30項目であるから、30枚のヒストグラムを描く必要がある。その時にヒストグラムを描く関数を繰り返し実行するためのコマンドが `for` 文である。`for(i in 1:30) {繰り返し実行する関数}` という形式である。

```
for( i in 1:30)  {
  hist(x.a[ , i], breaks=seq(0.5, 7.5, 1), xlab=names(x.a[i]), main="")
}
```

ヒストグラムを描く関数は `hist` 関数であり、そこでの分析対象の変数は、`x.a` のデータフレームの1列目から、30列目である。それを `x.a[, i]` という `i` で指定している。また7件法であるので、数値は 1, 2, 3, 4, 5, 6, 7 であり、0.5~1.5に何個、1.5~2.5に何個、・・・6.5~7.5に何個あるかを数えてヒストグラムにするために、`breaks` (分割) という引数を使う。`seq` 関数は、sequence (数列) の略である。`seq(0.5, 7.5,1)` というのは正確には、`seq(from=0.5, to=7.5, by=1)` ということであり、0.5, 1.5., 2.5, 3.5, 4.5, 5.5, 6.5, 7.5 という数列を出している。`xlab=names(x.a[i])` は、`x.a` のデータフレームの `i` 番目の変数名を、`x` 軸に表示する引数である。

さらに詳しく、項目それぞれの頻度表と割合を算出するためのスクリプトは、同じく `for` 文を使った下記のスクリプトとなる。ここでは、書き出すという意味の `print` 関数を使い、変数名を `names` 関数で、頻度表を `table` 関数で引き出している。また `nrow` 関数は `number of row` の略であり、行数を出している。

```
for(i in 1:30) {
  print(names(x.a[i]))
  print(table(x.a[i]))
  print(table(x.a[i])/nrow(x.a[i])*100)
}
```

あるいはより簡潔に、`apply` 関数を用いて下記のスクリプトでも実行できる。`apply` 関数について

は説明は省略する。

```
(count.a <- apply(x.a, 2, table))      #各列の頻度表
count.a/nrow(x.a)*100                #頻度割合
describe(x.a)                        #項目毎の基礎統計量
```

因子分析を行う際に、項目の天井効果、フロアー効果がある項目は削除したほうが良いとされている。その目安として mean+sd, mean-sd のチェックを行うとよい。これらが、解答選択肢の最小値である 1 を下回ったり、最大値である 7 を上回ったりした時には、その項目を削除したほうが良い場合がある。ただし、これらは機械的に行ってはならず、たとえ天井効果、フロアー効果があったとしても、項目内容からして、その項目が分析にとって重要で意味があるものならば、残して分析すべきである。

```
a <- sapply(x.a, mean)
b <- sapply(x.a, sd)
a+b
a-b
```

もし仮に第 1 項目と第 15 項目が、内容的にも不適切と判断したら、その項目を下記のように抜いて分析を続ければ良い。

```
x.a <- x.a[ -c(1, 15)]
names(x.a)
```

因子分析においては因子数を決定することが必要となるが、そのための決定的な基準というものは存在しない。いくつかの基準はあるが、あくまで項目内容と照らし合わせて、それぞれの因子に高い負荷量を持つ項目のかたまりが解釈可能となるように試行錯誤して因子数を決める必要がある。そのことを踏まえたうえで、下記の基準を参考にする。

```
# 因子数の検討
VSS.scree(x.a) # スクリープロット psychパッケージ カイザー基準とスクリー基準
```

```
fa.parallel(x.a) # 並行分析
vss(x.a)         # the Very Simple Structure (VSS) criterion
```

VSS.scree 関数では、スクリープロットが描かれる。固有値が1以上の因子数とするカイザー基準、また固有値の減衰が急激な箇所で因子数を決めるスクリー基準は、この関数で描かれるスクリープロットが参考になる。fa.parallel 関数では、並行分析という手法で推薦される主成分数と因子数が表示される。またvss 関数では、VSS 基準と呼ばれる結果が出される。この場合、表示された MAPが一番小さい値の因子数の採用が推薦となる。また、累積説明率によっても因子数の検討の参考になる。累積説明率が50%以上となるところで因子数を考えてもよい。

```
# 累積説明率の算出
e <- eigen(cor(x.a))
e$value                                # 回転前の固有値
cumsum(e$value)/sum(e$value)*100      # 累積説明率
```

因子分析を行う関数には factanal 関数が基本関数として組み込まれているが、この関数は最尤法しかサポートしていないため、psych パッケージにある fa 関数の使用を推薦する。指定しなければならないのは、因子分析の対象となるデータ、因子数、回転方法、因子分析手法の4つである。上記の手続きで3因子が適切だろうと推測したならば、nfactors=3 という引数で因子数を指定する。また回転には直交回転と斜交回転があり、代表的なものは、varimax 回転と promax 回転である。近年では、心理的概念が直交するとは考えにくいという理由で斜交回転が勧められているが、著者の体験では上手に作成された尺度項目なら varimax 回転で綺麗な単純構造の結果が出るので、varimax 回転も行うことを勧めたい。これらの指定は、rotate="promax" あるいは rotate="varimax" という引数で指定する。最後に手法であるが、factor method の略である fm の引数で指定する。数学的には最尤法 ml が優れているとされているが、場合によっては因子負荷量が1を超える値を出すような計算結果となるデータがある。そのような不適解となった場合には最尤法以外の方法を指定して、再度、因子分析を行う必要がある。私見であるが、最尤法 ml、重み付け最小2乗法 wls、主因子法 pa の順番で指定して実行することが良さそうである。なおオプションであるが、もし分析結果に基づく因子得点を保存して、後の分析に使うのであれば、scores=TRUE の引数をつける。下記は因子分析の結果を、fa.out というオブジェクトに納め、それを小数点以下3桁 (digit=3)、因子負荷量の高い順に並べ替えを指定 (sort=TRUE) の引数をつけて出力 print している。

```
# fa関数。3因子、プロマックス回転。最尤法
fa.out <- fa(x.a, nfactors=3, rotate="promax", fm="ml")
print(fa.out, digit=3, sort=T)          # sort=T で並び替えて表示
```

```
# プロマックス回転, 最小二乗法の場合
fa.out <- fa(x.a, nfactors=3, rotate="promax", fm="wls")
print(fa.out, digit=3, sort=T)
```

これらの因子分析表の結果を見て、因子負荷量がいずれの因子に対しても低い（たとえば .40以下）項目を削除したり、複数の因子に対して高い因子負荷量を持つ項目を削除したり、また因子数を変化させながら、適切な因子内容の解釈ができるまで因子分析を実行する。そして、適切だと解釈できる結果が得られたら、次に因子分析に基づいた、下位尺度を構成する。その際に、内的整合性である α 係数を下記のように検討する。たとえば、a2～a10までが一つの因子に、a11～a14, a16～a20が一つの因子に、a21～a30までが一つの因子に、それぞれ高い因子負荷量を示したとし、それぞれの項目から尺度を構成するとした場合の α 係数の算出方法は下記のとおりである。

```
#### 尺度作成のための内的整合性の検討
alpha(x[,c("a2","a3","a4","a5", "a6","a7", "a8", "a9", "a10")], check.keys=F)
alpha(x[, c("a11","a12","a13","a14","a16","a17","a18","a19","a20")], check.keys=F)
alpha(x[, c("a21","a22","a23","a24","a25","a26","a27","a28","a29","a30")], check.keys=F)
```

alpha 関数で引数, check.keys=F を追加しているが、これは逆転項目を自動的に逆転させて計算させないための措置である。もし、逆転項目があるならば、keys という引数で逆転項目を指定する。たとえば、a2, a6, a10 が逆転項目であるならば、下記となる。

```
alpha(x[, c("a2","a3","a4","a5", "a6","a7", "a8", "a9", "a10")], check.keys=F, keys=c("a2","a6","a10"))
```

alpha関数では、それぞれの項目を削除した時のアルファ係数の値が算出されるので、 α 係数を高めるために、それを参考に項目を削除していくことも考えられる。今回は、どの項目も削除したら α 係数が低くなると想定し、項目は削除せずに、高い因子負荷量を持つ項目から尺度得点を算出することとする。それぞれの尺度得点の列を、もともとのデータフレームである x に FA1, FA2, FA3 の変数名として追加することとする。尺度得点は単純合計を持って尺度得点とする場合もあるが、今回は、項目数で割った平均値を尺度得点とする場合を示す。

```
x$FA1 <- with(x, (a2+a3+a4+a5+a6+a7+a8+a9+a10)/9)
x$FA2 <- with(x, (a11+a12+a13+a14+a16+a17+a18+a19+a20)/9)
x$FA3 <- with(x, (a21+a22+a23+a24+a25+a26+a27+a28+a29+a30)/10)
```


以上の計算式は逆転項目がない場合である。もし逆転項目がある場合、例えば a06 得点が、1-7点ならば、 $(8 - a6)$ を加えることで逆転得点になるので、そのような修正をする必要がある。これらの尺度得点は項目数で割っているため、最低1から最高7までの得点分布になるので、下記のようにヒストグラムで分布をチェックすると良い。

```
# 尺度得点分布のチェック
hist(x$FA1, breaks=seq(0.5, 7.5, 1), xlab="FA1", main="")
hist(x$FA2, breaks=seq(0.5, 7.5, 1), xlab="FA2", main="")
hist(x$FA3, breaks=seq(0.5, 7.5, 1), xlab="FA3", main="")
```

```
#####
```

次に、b1-b20の2因子からなる既存の尺度の検討の方法である。調査対象が異なることで、尺度の因子が変化することを予想するならば、上記の探索的因子分析を実行して良い。既存の尺度であるため、先行研究と同じ因子構造であるかを確認するための、確認的因子分析を行っても良いだろう。下記に、確認的因子分析の実行の仕方を示す。この場合、lavaanパッケージのcfa関数を利用する。cfaはconfirmatory factor analysisの略である。

```
library(lavaan)
model.b <- '
facB1 =~ b1+b2+b3+b4+b5+b6+b7+b8+b9+b10
facB2 =~ b11+b12+b13+b14+b15+b16+b17+b18+b19+b20
'
```

lavaanパッケージをlibrary関数で呼び出す。次に、b1～b10が第1因子、b11～b20が第2因子に先行研究でわかれていたとした場合、model.bのオブジェクトに、そのモデルを上記のように指定する。その際、一重引用符で括弧することと、測定方程式を示す \equiv を使う。その上で、cfa関数を用いて、結果をfitに収納し、その結果の要約をsummary関数で表示する。その際に、引数、fit.measures=TRUE, standardized=TRUEも指定する。最後に、適合度を表示するfitmeasures関数を使う。

```
fit <- cfa(model.b, data=x)
summary(fit, fit.measures=TRUE, standardized=TRUE)
fitmeasures(fit)
```



```
#### 尺度作成のための内的整合性の検討
alpha(x[, c("b1","b2","b3","b4","b5","b6","b7","b8","b9","b10")], check.keys=F)
alpha(x[, c("b11","b12","b13","b14","b15","b16","b17","b18","b19","b20")], check.keys=F)
```

```
# 尺度の算出
```

```
x$FB1 <- with(x, (b1+b2+b3+b4+b5+b6+b7+b8+b9+b10)/10)
x$FB2 <- with(x, (b11+b12+b13+b14+b15+b16+b17+b18+b19+b20)/10)
```

```
#####
```

終わりに

以上で、R を利用した因子分析の実行と信頼性係数の検討も含めた尺度構成ができるようになるだろう。なお本稿で確認した R のバージョンは、3.1.2 であり、psych パッケージ (Revelle, 2015) のバージョンは 1.5.6、そして lavaan パッケージ (Yves, 2012) のバージョンは 0.5-18 である。これらは citation 関数または、sessionInfo 関数で調べることができる。またここでは説明していないがシンプルでわかりやすい資料としては、小杉・押江 (2013) があるので、そちらも参照してもらいたい。

文献

緒賀郷志 (2015). 心理調査におけるパス解析のための R スクリプト 岐阜大学教育学部研究報告 64(1) 59-65

小杉考司・押江隆 (2013). 中国四国心理学会第69回大会特別企画 Rチュートリアルセミナー特別テキスト
http://psycho.edu.yamaguchi-u.ac.jp/wordpress/wp-content/uploads/2014/01/R_tutorial20131.pdf

R Core Team (2014). R: A language and environment for statistical computing. R
Foundation for Statistical Computing, Vienna, Austria. URL <http://www.R-project.org/>.

Revelle, W. (2015). psych: Procedures for Personality and Psychological Research, Northwestern
University, Evanston, Illinois, USA, <http://CRAN.R-project.org/package=psych> Version = 1.5.6.

Yves Rosseel (2012). lavaan: An R Package for Structural Equation Modeling.
Journal of Statistical Software, 48(2), 1-36. URL <http://www.jstatsoft.org/v48/i02/>.