

## 18日目：相関係数（3）

さて、今日は論文掲載用の表に加工しやすい出力を目指した自作関数について紹介します。題材に、緒賀先生が作られている関数「simple.car」を取り上げてみようと思います。

この関数は、以下のサイトで紹介されているので、まずはそこを見てください。

<https://sites.google.com/site/officeoga/r/memo>

使用例をみるとわかりやすいと思いますが、この関数の考え方（出力目標）は、相関係数のマトリックス、無相関検定結果(p値)のマトリックス、その結果を記号表記したマトリックスを並べて表示できるようにしようというものです。

なお、使用例では、`simple.cor(iris[c(1,3,4,2)])`という命令がされていますが、これは、`iris`というデータの、1, 3, 4, 2列目を使うという指示です。xの2行目から9行目ならば、`simple.cor(x[2:9])`とすればOK。

このような表示を目指した関数を丁寧に(?)説明してみると…

```
simple.cor <- function(x)
  { a <- x
    nc <- ncol(a)
```

function は、これから関数を定義しますという命令。その関数の名前は <- で示された simple.cor。関数の中身は、aの前の「{」から、最後のまで「}」  
作業のために指定されたデータ x を a という名前にしておき、a の変数の数（列数）を調べ nc に代入します。

```
mn <- rep(NA, nc*nc)
mnr <- matrix(mn, ncol=nc)
pm <- matrix(mn, ncol=nc)
pmd <- matrix(mn, ncol=nc)
```

ここでは結果を表示するマトリックス（入れ物）を用意しています。まず準備として、そのマトリックスに必要なセルの数だけ NA を並べた変数リスト mn をつくっておく（変数の数が 4 なら、4×4 の 16 のセルが必要になるので、nc\*nc 個の NA を作成）。この mn を、変数の数(nc)を列数とするマトリックスに変更。mnr という名前で相関係数表示用。pm という名前で無相関検定の p 値表示用。pmd という名前で無相関検定の簡易表示用に利用します。

```
colnames(mnr) <- colnames(a)
rownames(mnr) <- colnames(a)
colnames(pm) <- colnames(a)
rownames(pm) <- colnames(a)
colnames(pmd) <- colnames(a)
rownames(pmd) <- colnames(a)
```

ここは、元のデータ(a)の変数名を、結果を入れるマトリックス(mnr, pm, pmd)の行および列につけています。

次が cor.test を活用した計算部分になるのですが、それを読み取る前に…

cor.test の結果は一つのオブジェクトであり、いくつかの内容が含まれています。このオブジェクト内にどのような情報があるかを確認してみます。

```
cor.test(d$a1, d$a2) -> a1.a2
```

```
names(a1.a2)
```

すると、a1.a2には、以下のような9の変数が含まれていることがわかります。

```
[1] "statistic" "parameter" "p.value" "estimate" "null.value" "alternative"
```

```
[7] "method" "data.name" "conf.int"
```

以下の計算では、これらから estimate(相関係数)、p.value(p 値)を抽出して表示に使っています。

以下で for(m in 1:nc) {} というのは、「{} 中の m の値を 1 から nc まで1ずつ順に動かせ」という命令です。これが m と n の 2 つについて指示されているので、(m,n)は、(1,1),(1,2),(1,3)…(1,nc), (2,1),(2,2),… (2,nc), (3,1),(3,2)… (3,nc)… …(nc,nc) と順に動かすこととなります。

ただし if(m==n) next とあるように、m=n の場合(つまり同一変数間)は飛ばします。

```
for(m in 1:nc)
```

```
  for(n in 1:nc) {
```

```
    if(m==n) next
```

```
    r <- cor.test(a[,m],a[,n])$estimate
```

```
    mnr[m,n] <- round(r,3)
```

```
    p <- cor.test(a[,m],a[,n])$p.value
```

```
    pm[m,n] <- round(p,3)
```

```
    pmd[m,n] <- ifelse(p<=.001,"***",ifelse(p<=.01,"**",ifelse(p<=.05,"*", "---")))
```

for により、まず、a の(m,n)に、(1,1)を入れて…、とはなりません。m=n の場合は飛ばすということでしたから、最初は(1,2)を入れます。そして cor.test を実施し、結果の中の estimate の値を r に入れます。その r を 3 桁で丸めて、mnr の[1,2]に入れます。

同様に、結果の中の p.value の値を p に入れ、その値を 3 桁で丸めて、pm の[1,2]に入れます。

ifelse は、「もし●が真ならば△せよ、●が真でなければ(偽であれば)×せよ」というもの。つづくカッコの中は3つに分けられていて、最初が●、2番目が△、3番目が×に対応しています。わかりやすいのは、ifelse(p<=.05,"\*", "---")でしょう。p について p<=.05 が真であれば「\*」を、偽であれば「---」を指示された pmd[1,2]に入れるということになります。その前の2つの ifelse は、偽の部分にさらに ifelse が入っているという形式です。

これを最終セルの(nc,nc)まで続けます。

```
cat("相関係数","¥n" )
print(mnr)
cat("\n")
cat("無相関検定結果の p-value","¥n")
print(pm)
cat("\n")
cat("無相関検定結果の簡易表示","¥n")
print(pmd, quote=F)
cat("** p<.05, ** p<.01, *** p<.001","¥n")
}
```

ここが結果表示をコントロールする部分です。  
cat は、次の"" 内の文字列を表示させる命令で、¥n は改行せよという命令。  
print は、それを表示せよという命令。ここで結果が入った各マトリックスを指定します。  
pmd の部分は文字列表示なので、quote=F をつけないと "\*\*\*" のように "" が付いてしまいます。

このような自作関数は、web 上にたくさん公開されているので、試してみる価値はあるでしょう。ただし、結果の正しさは保証されるものではありませんので、ちゃんと自分で確認しておくことが必要でしょう。

これら関数の使い方は、まず web の関数部分をコピーし、R コンソールか R エディタに貼り付けます。

R コンソールにペーストした場合には、その後エンターキーを押します。エンターを押すと、各行頭に「+」という記号が付されると思います。こうなれば、simple.cor という関数が使えるようになります。

また、R のパッケージのように必要な時に呼び出す使い方もできます。そのためには、新しい R エディタファイルを開き、関数をペーストします。そして名前をつけて保存しておきます。その関数を使いたい時には、R のメニューバーの「ファイル」から、「R コードのソースを読み込み」を選びます。すると読み込む R ファイルを尋ねてきますので、関数を保存したファイルを指定して開きます。

これを行うと、R コンソールには

```
> source(" ..... 関数を保存したファイル")
```

というようにそのファイルの位置と名前が表示されます。これで、その関数を使えるようになります。実際には、このやり方が一番便利かもしれません。

結果を表にすると以下のような感じになります。

	親の様子	家にある本	知識欲	不可欠さ	インターネット	マンガ	本好き	読書習慣
親の様子	---	.687 ***	.288 ***	.325 ***	.150 ns	.044 ns	.475 ***	.493 ***
家にある本		---	.340 ***	.347 ***	-.058 ns	-.046 ns	.321 ***	.422 ***
知識欲			---	.549 ***	.275 ***	.312 ***	.223 **	.395 ***
不可欠さ				---	.034 ns	.172 *	.246 **	.234 **
インターネット					---	.301 ***	.237 **	-.030 ns
マンガ						---	.025 ns	.006 ns
本好き							---	.547 ***
読書習慣								---

さて、このような緒賀先生のプログラムはとても便利なのですが、個人的にはもう少し要望もあります。①論文記載の表は、相関係数と結果表示記号がとなりになっている場合が多いので、別の表ではなく、ひとつの表として処理できるように表示したいこと。②自分の作った関数を信用しないわけではないけれども、結果記号が合っているかどうかを確認するために、p値の一覧があること。③欠損値のあるデータを確認するために、人数も表にしたいこと。

そんなこんなで、緒賀先生のものをお手本に、少し修正を加えて自作してみました（`simple.cor2` という関数名。`simple.cor2(x[2:9])`などと指定してやればOK）。

# 以下浦上修正の関数

```
simple.cor2 <- function(x)
  { a <- x
    nc <- ncol(a)
    mn <- rep(NA, nc*nc)
    pm <- matrix(mn, ncol=nc)
    pmd <- matrix(mn, ncol=nc)
    mn2 <- rep(NA, nc*nc*2)
    allm <- matrix(mn2, ncol=2*nc)
    ns <- matrix(mn, ncol=nc)
    colnames(pm) <- colnames(a)
    rownames(pm) <- colnames(a)
    rownames(allm) <- colnames(a)
    colnames(ns) <- colnames(a)
    rownames(ns) <- colnames(a)

    for(m in 1:nc)
      for(n in 1:nc) {
        if(m==n) next
```

```
r <- cor.test(a[,m],a[,n])$estimate
allm[m,2*n-1] <- round(r,3)
p <- cor.test(a[,m],a[,n])$p.value
pm[m,n] <- round(p,3)
allm[m,2*n] <- ifelse(p<.001,"***",ifelse(p<.01,"**",ifelse(p<.05,"*", "ns")))
ns[m,n] <- sum(!is.na(a[,m] & a[,n])) # !is.naはNAを除くという命令
}
```

```
cat("相関係数総合表示","\n ")
print(allm, quote=F)
cat("** p<.05, ** p<.01, *** p<.001","\n")
cat("\n")
cat("無相関検定結果のp-value","\n")
print(pm)
cat("\n")
cat("人数","\n")
print(ns)
cat("\n")
}
```

これは Mac 用には書いてあるので、Win なら表示のコントロール部分の `\n` は、`¥n`（¥は半角）の方が入力しやすいと思います。

出力結果は自分で確認してください、ということなのですが、まあ、イメージに近いものを得ることができているような気がします（美しくないけど…）。

自作関数が扱える（作れる）ようになると、自動化できる部分も増え、分析（というか、結果の加工過程？）も楽になるのではないかと思います。

本日はここまでです。