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，「{ } の中の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) と順に動かすことになります。
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。なお，web では，￥n になっていますが，Mac だと表示しにくいので，ここでは同じ意味になる \n を使っています（「\」は，delete キーの横にある「￥」キーで入力できます）
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。

R エディタの場合は，一連のことをまとめて実行できますので，続けて simple.cor(x[5:10])などという命令も書いておいて，一括して実行できます。なお，R エディタにペーストする場合には，単にペーストするのではなく，「テキストとしてペースト」を選んでおくことをすすめます。普通にペーストするとエラーが出るけれども，「テキストとしてペースト」ならば問題なく動くということが結構あります。（個人的な経験ですが，コピペしてきた関数は，そのままでは動かないこともあります…。Mac ユーザーの悲しいところですが…。原因は改行コードあたりなのだろうなと思うのですが，エラーが出てきます。改行コードは見えない（見る方法があるのだろうか？）ので，対処がやっかいなのです。これは関数自体に問題があるのではないので，エラーが出たあたりを入力し直すなどやってみてください）

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，「ソースを読み込む」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("/Users/urakami　…… /関数を保存したファイル")
```

というように表示されます。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")
}
```

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。