

R言語解説

個体群行列モデルデータベース
COMPADRE・COMADRE使用のために

高田 壮則(北海道大学)

R およびR統合環境のインストール・起動

Rのインストール

- The R Project for statistical computing : <http://www.r-project.org/>から実行ファイルをダウンロードします。

R Studioのインストール


- いわゆるRのIDE（統合開発環境）
- 4つの画面にプログラムファイル、入力したコマンド履歴、実行コンソール、グラフ画面などが入りされています。
- Rのインストールとは別にインストールする必要があります。

手順1 : R studioのインストーラをダウンロードします

- <https://www.rstudio.com/products/rstudio/download/>
- 自分の使用しているOSにあわせてダウンロードします。

手順2 : ダウンロードしたファイルを開いてインストールをします

手順3 : インストールした後で、アプリケーションフォルダ（MacとWindowsでちょっと違う）に「R studio」があれば、インストール完了です。

手順4 : 起動のためにアイコン  をクリックする

作業ディレクトリの設定・変更 データの入出力

```
getwd() # 現在の作業ディレクトリの確認
[1] "/Users/?????????????" # Rからの結果の出力
setwd("/Users/????????????? /Desktop") # 作業ディレクトリを変更,でも反応なし
getwd() # 作業ディレクトリの再確認
```

(注) #記号のあとは、コメント文（自分のためのメモ）をつけても構いません。

以下は後ほど使います

```
load("ファイル名") # ファイルを読み込む。Rdataの時は、ファイル名の
                   # 最後に拡張子.Rdataが付いています。
write.csv(*****, file = "metadata.csv",fileEncoding = "UTF-8")
# ****の内容をmetadata.csvというファイルにcsv形式で書き出す
```

データフレーム・構成の確認

* データフレームとは R のオブジェクトの型の一つ.

	SEX	HEIGHT	WEIGHT
1	F	158	51
2	F	162	55
3	M	177	72
4	M	173	57
5	M	166	64

各行・列はラベルを必ず持つ.
ラベルを使って操作できる。

```
sex <- c("F","F","M","M","M") #ベクトルを作る
height <- c(158,162,177,173,166)
weight <- c(51,55,72,57,64)
x <- data.frame(SEX=sex, HEIGHT=height, WEIGHT=weight) #ラベルを使った操作
```

```
x$HEIGHT      #xというリストのWEIGHT列の中身を見る
x[[1,2]]       #要素を見ている
names(x)       # x の列名を表示
colnames(x)   # xの列名を表示
head(x, n=a)   # 先頭から a 行だけ抽出する
```

```
rbind(x, y)    # x と y を"縦に並べて結合する (x と y の列名が全て同じ場合) .
cbind(x, y)    # x と y を横に並べて結合する (x と y の行数が同じ場合) .
rowSums()      行の総和
colSums()      列の総和
```

Data の読み込み・構成の確認・出力

入力

```
x <- read.table("data01.txt") # data01.txtをxという名前のデータフレームとして読み込む
x <- read.table("data01.txt", header=T, sep=",")
# 同じことを1行目に列名が入っているデータで区切り記号が","の場合に
x <- read.csv("ファイル名") # csvファイルとして読み込む
```

など、様々なオプションがある。読み込んだ時に名前を与えておいて、その名前を使って内容を確認したり、計算をしたりする。

COMPADREデータベースの場合は、
`load("/Users/ユーザー名/Desktop/COMPADRE_v.6.21.1/COMPADRE_v.6.21.1.0.RData")`

出力

```
print(x) # xの内容をコンソール上に出力
write.csv(x, file = "*****.csv", fileEncoding = "UTF-8")
# xの内容を*****.csvというファイルにcsv形式で書き出す
write(リスト名, "出力するファイル名", append = T, ncolumns = 一行の要素の数)
write.table(データフレーム名, "出力するファイル名", append=T, col.names=F)
```

関数

\sqrt{a}	<code>sqrt(a)</code>
a^b	<code>a**b</code>
$ a $ (note: for complex arguments, this computes the modulus)	<code>abs(a)</code>
e^a	<code>exp(a)</code>
$\ln(a)$	<code>log(a)</code>
$\log_2(a)$, $\log_{10}(a)$	<code>log2(a)</code> , <code>log10(a)</code>
$\sin(a)$, $\cos(a)$, $\tan(a)$	<code>sin(a)</code> , <code>cos(a)</code> , <code>tan(a)</code>
Round to nearest integer	<code>round(x)</code>
Round down to next lowest integer	<code>floor(x)</code>
Round up to next largest integer	<code>ceiling(x)</code>
Sign of x (+1, 0, or -1)	<code>sign(x)</code> (Note: for complex values, this computes <code>x/abs(x)</code> .)
Obtaining average	<code>mean(x)</code>
Obtaining maximum	<code>max(x)</code>
Summing the elements of x	<code>sum(x)</code>

ベクトル・行列

ベクトル

```
ec <- c(1.0, 2.0, 3.0, 4.0, 5.0)      # ec という名のオブジェクトにベクトルを代入
(ec <- c(1, 2))                       # 丸括弧で囲むと、代入した結果を表示
length(ec)                            # ベクトルの長さを調べる
ec[k] <- a                             k 番目の要素を a に変更.
```

行列

```
(x <- matrix(c(1,0.5,3,0,0.2,4), nrow=2, ncol=3)) # 行列を作る
```

x[2,]	2 行目を取り出す	x[,2]	2 列目を取り出す
x[1, 2]	1 行 2 列目の成分を取り出す	dim(x)	行数、列数を調べる

a * b	# 要素ごとの掛け算	a %*% b	# 行列の掛け算
diag(3)	# 3 × 3 の単位行列を作る.		
diag(1:3)	# 対角成分が (1 2 3) の 3 × 3 対角行列を作る		
t(X)	行列 X を転置		
solve(X)	行列 X の逆行列		
eigen(X)	行列 X の固有値と固有ベクトルを求める.		
det(X)	行列 X の行列式		

繰り返し操作

for (i in a:b) {繰り返す内容（複数の命令が入ってもいい）
i をa:bのリストから順に拾いb-a+1回繰り返す

例 1
繰り返す内容
がiの値に無関
係

```
x <- 0
for (i in 1:5) {
  x <- x + 1
}
x
```

for (ループ変数 in ベクトル)
ベクトルの要素が空にならない限り式が繰り返される
xをコンソールに出力

例 2
繰り返す内容
がiの値に依存
する

```
x <- 0
for (i in 1:5) {
  x <- x + i
}
x
```

for (ループ変数 in ベクトル)

例 3
複数の命令が
行われる

```
for (i in 1:3) {
  s <- matrix(1:4, nrow=2, ncol=2)
  print(eigen(s+i))
}
```

行列 $\begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix}$ を作る
行列 $\begin{pmatrix} 1+i & 3+i \\ 2+i & 4+i \end{pmatrix}$ の固有値・固有ベクトルの出力

条件分岐

ある条件で場合分けをして処理を行いたい場合は if 文, else 文を使う.

`if (条件式) { 命令 1 } else 命令 2 }` 条件がTRUEなら命令 1, FALSEなら命令 2

条件式	<code>x < a</code>	True if x is less than a
	<code>x > a</code>	True if x is greater than a
	<code>x <= a</code>	True if x is less than or equal to a
	<code>x >= a</code>	True if x is greater than or equal to a
	<code>x == a</code>	True if x is equal to a
	<code>x != a</code>	True if x is not equal to a

条件が 2 個ある場合

- | | |
|-----------------|--------------|
| (1) 論理積 (かつ条件) | 条件 1 && 条件 2 |
| (2) 論理和 (または条件) | 条件 1 条件 2 |

例 1

```
for (i in 1:5) {  
  s <- matrix(1:4, nrow=2, ncol=2)  
  if(i<3) print(eigen(s+i)) else print(sum(s))  
}
```

もし i が 1, 2 なら固有値・固有ベクトルを出力し、
 i が 3, 4, 5 なら全要素の和を出力する

ちょっと遊び

```
ke<-c()  
for (i in 1:5) {  
  s <- matrix(1:4, nrow=2, ncol=2)  
  ans<-eigen(s+i)  
  ke[[i]]<-ans$value[[1]]  
}
```

適当に変化をつけてみよう

