データ構造と操作

第 *2* 講 - *R* 言語で扱うデータとその演算

村田 昇

講義概要

- R 言語のデータ構造
- ベクトル・行列・リストの操作
- ベクトルと行列のさまざまな計算

R 言語のデータ構造

基本的なデータ構造

- 下記は基本的なもので標準環境で利用できる
  - ベクトル (vector)
  - 行列 (matrix)
  - リスト (list)
  - データフレーム (data frame)
  - 配列 (array) (今回は扱わない)
- パッケージなどで拡張することができる
  - データフレームの拡張 (`data.table` など)
  - 時系列の扱い (`zoo`,`xts` など)

ベクトル

ベクトルとは

- スカラー値 (単一の値) の集合
- スカラー値として扱われる主なデータ型
  - 数値 (実数や複素数)
  - 文字列 (' や "で囲まれた文字．"foo"，"bar"など)
  - 論理値 (`TRUE` , `FALSE`)
- R オブジェクトの多くはベクトルとして扱われる
- スカラーは長さ 1 のベクトルとして扱われる

ベクトルの生成と操作

- 基本的には関数 `c()` を用いて作成する
- 数値や文字列の要素からなるベクトルの生成

```
(x <- c("Alice","Bob","Cathy","David")) # 文字列のベクトル
(y <- c(1,-2,3,-4,5)) # 数値のベクトル
(z <- c("apple","berry","cat","dog","elephant")) # 文字列のベクトル
## 外側の () は代入した結果の表示．print() と同義
```

- ベクトルの長さの取得 (関数 `length()`)

```
length(x) # 最後の要素を参照する場合などに利用できる
```

- ベクトルの要素の取得 (演算子 `[]`)

```
x[3] # x の第 3 要素 (ベクトルの添え字は 1 から始まる)
y[c(1,3,4)] # 複数の要素 = c(y[1],y[3],y[4])
```

- a から b まで 1 ずつ変化するベクトル (演算子 `:`)

```
a <- 8; b <- 15 # 変数 a,b に値を代入．複数コマンドは ; で区切る
a:b # a < b の場合は 1 ずつ増加する系列が作成される
a <- 29.5; b <- 24
a:b # 逆の場合は 1 ずつ減少する系列が作成される
29.5:24 # 直接数値を書いてもよい
```

- a から b まで c ずつ変化するベクトル (関数 `seq()`)

```
a <- 1; b <- 16; c <- 2 # 変数 a,b,c に値を代入
seq(a,b,by=c) # 明示する場合は seq(from=a,to=b,by=c)
```

- ベクトルの繰り返し (関数 `rep()`)

```
rep(y,3) # 長さは length(y) * 3
rep(y,times=3) # 単純に繰り返す．上記と同様
rep(y,each=3)  # 各要素を繰り返す
rep(y,length.out=12)  # 繰り返した結果の長さを指定する
```

- ベクトルの反転 (関数 `rev()`)

```
rev(x)
```

- ベクトルの結合 (関数 `c()`)

```
c(x,z) # 同じデータ型のもの単純に結合される
c(x,y) # 異なるデータ型のものは結合できないので自動的に書き換えられる
```

演習

練習問題

ベクトルの操作に慣れよう

- 以下に示すベクトルを作成してみよう
  - 1 から 10 までの自然数のベクトル
  - 1 以上 30 以下の奇数を昇順に並べたベクトル
  - すべての要素が 1 からなる長さ 10 のベクトル
- 作成したベクトルを操作してみよう
  - ベクトルの長さを求める
  - 3 番目の要素を取り出す
  - 最後の要素を取り出す

行列

行列とは

- スカラー値を 2 次元状 (縦横) に並べたもの
  - 縦 (列) ベクトルを列方向に並べて束ねたもの
  - 横 (行) ベクトルを行方向に並べて束ねたもの
- データ型は何でもよいが **混在はできない**
- データフレームは行列を拡張したもの

行列の生成と操作

- すべての要素が a である $m \times n$ 行列の生成
  (関数 matrix())

```
a <- 2; m <- 3; n <- 4 # 変数 a,m,n に値を代入
matrix(a,m,n) # 明示する場合は matrix(data=a, nrow=m, ncol=n)
```

- 長さ mn のベクトル a を $m \times n$ 行列に変換
  (関数 matrix())

```
a <- 2:13 # 変数 a に値を代入 (m*n = 12 文字用意)
(A <- matrix(a,m,n)) # 並び順を変えるには matrix(a,m,n,byrow=TRUE)
```

- 行列のベクトル化 (関数 as.vector())

```
as.vector(A) # matrix の逆変換にあたる
```

- 長さが等しい複数のベクトルの結合
  (関数 rbind(),cbind())

```
a <- 4:7; b <- 10:7; c <- c(2,4,8,16) # 変数 a,b,c に値を代入
rbind(a,b,c) # 行ベクトルとして結合 (row vector bind)
cbind(a,b,c) # 列ベクトルとして結合 (column vector bind)
```

- 行列のサイズの取得 (関数 dim() とその仲間)

```
dim(A) # 長さ 2 のベクトル (行数, 列数) となることに注意
nrow(A) # 行数
ncol(A) # 列数
dim(A)[1] # nrow(A) と同値
dim(A)[2] # ncol(A) と同値
```

- 行列の成分の取得 (演算子 [])

```
A[3,4] # (3,4) 成分
A[3, ] # 第 3 行のベクトル
A[ ,4] # 第 4 列のベクトル
A[c(1,3),] # 1,3 行からなる部分行列
A[c(1,3),2:4] # 1,3 行と, 2,3,4 列からなる部分行列
```

行列の操作に関する補足

- 関数 cbind()/rbind() は行数・列数が等しい行列も横・縦に結合できる
- 行列の高次元版として配列 (array) が用意されている
- 関数 rownames()/colnames() を用いると行と列に名前を付けることができる
- これらの機能は講義の中で使いながら説明する

演習

練習問題

行列の操作に慣れよう

- 以下に示す行列を作成してみよう

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

- 行列を操作してみよう
  - 2 行 2 列成分を取り出す
  - 転置行列を作成する
  - 行名をつける

その他のデータ構造

リストとは

- 異なる構造のオブジェクトを 1 つにまとめたもの
  - リストの各要素は異なるデータ型・サイズであって構わない
  - 複数のデータフレームを扱うときなどに応用可能
- 本講義のデータ解析ではほとんど用いない
  - R の関数の操作ではときどき必要

リストの生成と操作

- リストの生成 (関数 `list()`)

```
(L <- list(x,y)) # x,y を要素とするリスト
```

- リストの要素の参照 (演算子 `[[]]`)

```
L[[1]] # リストの第 1 要素
```

- リストの各要素に名前を付与 (関数 `names()`)

```
## 方法 1 (作成時に名前を付与)
(L1 <- list(first=x, second=y))
```

```
## 方法 2 (作成後に名前を変更)
L2 <- list(x,y)
names(L2) <- c("1st","2nd")
L2 # リストを表示
```

- 名前によるリストの要素の取得 (演算子 `[[]]`,`$`)

```
## 方法 1 (リストの名前で参照)
L2[["1st"]]
```

```
## 方法 2 (記号$を用いる場合は""は不要なことに注意)
L1$first
```

データフレームとは

- 長さの等しいベクトルを束ねたリスト
- 複数の属性を持つ実データに則したデータ構造
- 各列は異なるデータ型でも良い
- データフレームは **リスト** でもある
  リストと同様にして各変数を取得できる
- データフレームは **行列** のように扱える
  行列と同様にして各変数を取得できる

データフレームの生成

- データフレームの生成 (関数 `data.frame()`)

```
## 前回の練習問題の例
df <- data.frame(
    literature=c(90,80,70,60),
    math=c(25,50,75,100),
    english=c(65,100,70,40))
row.names(df) <- x # 各行の名前を書き換える
df
```

- 操作については別の講義で詳細に説明する

ベクトルの計算

ベクトルの表記

- ベクトルは太字で，要素は下付き添字で表す

$$\boldsymbol{a} = (a_1, a_2, \ldots, a_k)$$

- 別の書き方

$$(\boldsymbol{a})_i = (\text{ベクトル } \boldsymbol{a} \text{ の第 } i \text{ 成分})$$

- R の書式 (関数 `c()`)

```
a <- c(a1,a2,...,ak) # k次元ベクトルの作成 (擬似コード)
```

ベクトルの加法

- **同じ長さのベクトル** の和および差
  数値の和と差のように扱うことができる

$$\boldsymbol{a} \pm \boldsymbol{b} = (a_1 \pm b_1, a_2 \pm b_2, \ldots, a_k \pm b_k)$$

$$(\boldsymbol{a} \pm \boldsymbol{b})_i = a_i \pm b_i$$

- R の書式 (演算子 `+`, `-`)

```
a + b # 同じ長さのベクトル a,b の和．同じ長さのベクトルが返る
a - b # ベクトルの差
```

ベクトルの乗法

- **同じ長さの 2 つのベクトル** の乗法
  - 成分ごとの積 (Hadamard 積; 要素積)
  - ベクトルの内積

  2 種類あることに注意する
- データ解析ではどちらも良く用いられる

Hadamard 積

- **同じ長さのベクトル** の成分ごとの積

$$\boldsymbol{a} \circ \boldsymbol{b} = (a_1 b_1, a_2 b_2, \ldots, a_k b_k)$$

$$(\boldsymbol{a} \circ \boldsymbol{b})_i = a_i b_i$$

- R の書式 (演算子 `*`, `/`)

```
a * b # ベクトルの成分ごとの積．同じ長さのベクトルが返る
a / b # 成分ごとの商も計算可
```

内積

- **同じ長さのベクトル** の内積

$$\begin{aligned}\boldsymbol{a} \cdot \boldsymbol{b} &= a_1 b_1 + a_2 b_2 + \cdots + a_k b_k \\ &= \sum_{i=1}^{k} a_i b_i\end{aligned}$$

- R の書式 (演算子 `%*%`)

```
a %*% b # ベクトルの内積．1x1 型の行列が返る
```

行列の計算

行列の表記

- 行列は大文字で，要素は下付き添字で表す

$$A = \begin{pmatrix} a_{11} & \ldots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \ldots & a_{mn} \end{pmatrix}$$

- 別の書き方

$$(A)_{ij} = (\text{行列 } A \text{ の } ij \text{ 成分})$$

- R の書式 (関数 `matrix()`)

```
A <- matrix(c(a11,a21,...,amn),m,n) # m x n 行列の作成 (擬似コード)
```

行列の加法

- **同じ大きさの行列** の和および差
  ベクトルと同じように記述することができる

$$(A \pm B)_{ij} = a_{ij} \pm b_{ij}$$

- R の書式

```
A + B # 同じサイズの行列の和．同じサイズの行列が返る
A - B # 行列の差
```

行列の乗法

- 2 つの行列の乗法
  - **同じ大きさの行列** の成分ごとの積 (Hadamard 積; 要素積)
  - $n \times m$ **型行列と** $m \times l$ **型行列** の積

  2 種類あることに注意する
- データ解析ではどちらも良く用いられる

Hadamard 積

- **同じ大きさの行列** の成分ごとの積

$$(A \circ B)_{ij} = a_{ij} b_{ij}$$

- R の書式 (演算子 `*`, `/`)

```
A * B # 行列の成分ごとの積．同じサイズの行列が返る
A / B # 成分ごとの商も計算可
```

行列の積

- $n \times m$ **型行列** A **と** $m \times l$ **型行列** B の積

$$(AB)_{ij} = \sum_{k=1}^{m} a_{ik} b_{kj} \quad (AB \text{ は } n \times l \text{ 行列})$$

- R の書式 (演算子 `%*%`)

```
A %*% B # 行列の積．n x l 型行列が返る
```

行列式

- n **次正方行列** A の行列式

$$\det(A) \quad (= |A|)$$

正方行列でなければ定義されないことに注意する

- R の書式 (関数 `det()`)

```
det(A) # 行列式
```

トレース

- n **次正方行列** A のトレース (対角成分の総和)

$$\mathrm{trace}(A) = \sum_{i=1}^{n} a_{ii}$$

- R の書式 (関数は用意されていないので以下を利用)

```r
sum(diag(A)) # 行列のトレース
```

- 関数 `diag()`：行列の対角成分を取り出す
  (ベクトルを対角行列にすることもできる)
- 関数 `sum()`：ベクトルの総和を計算する

演習

例題

- 適当な 2 次正方行列 A で Hamilton-Cayley の定理

$$A^2 - \mathrm{trace}(A)A + \det(A)E_2 = O_2$$

の成立を確認せよ
ただし E_2 は 2 次単位行列，O_2 は 2 次正方零行列

解答例

```r
## 行列を作成（好きに設定してよい）
(A <- matrix(1:4,2,2) - diag(rep(3,2)))
## 左辺を計算（丸め誤差の範囲で 0 になる）
A %*% A - sum(diag(A)) * A + det(A) * diag(rep(1,2))
```

練習問題

- 1 から 10 の 2 乗値からなるベクトルを作成せよ
- 1 から 10 までの和を計算せよ
- 行列を用いて九九の表を作成せよ
- 30 度の回転行列を 2 回乗ずると 60 度の回転行列となることを確認せよ

$$(\text{回転行列}) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$$

ベクトルと行列の計算

ベクトルと行列の乗法

- 列 (縦) ベクトル・行 (横) ベクトルという **区別はない**
- 行列とベクトルの順序で適切に判断される
- 計算結果は **行列** で表現される (演算子 `%*%`)

```r
A <- matrix(1:4,2,2); b <- c(5,6) # 行列とベクトルを作成
A %*% b # 行列 x ベクトル = 列ベクトル
```

```r
b %*% A # ベクトル x 行列 = 行ベクトル
```

連立 1 次方程式の解法

- 連立 1 次方程式
  - A : n 次正則行列
  - $\boldsymbol{b}, \boldsymbol{x}$: n 次元列ベクトル

$$A\boldsymbol{x} = \boldsymbol{b} \qquad \text{(連立 1 次方程式)}$$
$$\boldsymbol{x} = A^{-1}\boldsymbol{b} \qquad (A \text{ が正則な場合})$$

- 解を求めるには関数 `solve()` を利用する

```
x <- solve(A, b)
```

- ベクトル $\boldsymbol{b}$ の代わりに行列も扱える

逆行列

- 正則な n 次正方行列 A の逆行列 A^{-1}

$$AA^{-1} = A^{-1}A = E_n \quad (E_n \text{ は } n \text{ 次単位行列})$$

- 関数 `solve()` を利用して求めることができる

$$AX = E_n$$
$$X = A^{-1}E_n = A^{-1}$$

```
solve(A,B) # AX=B の解 X を求める
solve(A) # 逆行列 (B が単位行列の場合省略できる)
```

- 他にもいくつか方法は用意されている

関数の適用

- ベクトルや行列に関数 ($\sin, \exp, \ldots$ など) を適用すると成分ごとに計算した結果が返される
- ベクトル $\boldsymbol{a}$, 行列 A に関数 sin を適用する

$$(\sin(\boldsymbol{a}))_i = \sin(a_i)$$

```
sin(a) # 成分ごとに計算される. sin(a)[i]=sin(a[i])
```

$$(\sin(A))_{ij} = \sin(a_{ij})$$

```
sin(A) # 成分ごとに計算される. sin(A)[i,j]=sin(A[i,j])
```

演習

例題

- 適当な 3 次正方行列 A と 3 次元ベクトル $\boldsymbol{b}$ を作成して $\boldsymbol{x}$ に関する以下の連立 1 次方程式を解きなさい

$$A\boldsymbol{x} = \boldsymbol{b}$$

解答例

```
(A <- matrix(rnorm(9),3,3)+diag(rep(1,3))) # 行列とベクトルを作成
## rnorm(9) は正規乱数を 9 つ作成する (後の講義で詳しく説明)
(b <- 1:3)
```

```
(x <- solve(A,b)) # 解を計算
A %*% x # 結果の確認 (b になるはず)
```

練習問題

- 1 から 10 の 2 乗値からなるベクトルを作成せよ
- 例題の A と $\boldsymbol{b}$ を用いて

  ```
  A %*% b + b %*% A
  ```

  を計算するとエラーになるが，何故そうなるか理由を考えよ
- 2 次元ベクトルを回転行列で変換しても長さが変わらないことを確かめよ

次回の予定

- R 言語における関数
- 引数の扱い方 (引数名・順序・既定値)
- 自作関数の定義
- 制御構造 (条件分岐・繰り返し)