Rで学ぶプログラミングの基礎の基礎

(8) 複雑なくり返し文

本日のメニュー

1. 行列について
2. 複雑なくり返し文

行列の作成

- 行列 $\begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix}$ を作成する手順は以下のとおり

1. 行列の要素をベクトルで用意する
2. 関数 matrix(ベクトル, 行数, 列数) でベクトルから行列に変換する

```
> x <- c(1, 2, 3, 4, 5, 6)
> A <- matrix(x, 2, 3)
> A
     [,1] [,2] [,3]
[1,]    1    3    5   ← 1 行目
[2,]    2    4    6   ← 2 行目
```

1列目　2列目　3列目

行列計算

演算子

演算子	＋	－	%*%	*
意　味	加算	減算	行列の積	対応する要素同士の積

数学関数

関数	機能
chol(A)	行列 A のコレスキー分解を行う
crossprod(A)	行列 A のクロス積を求める
eigen(A)	行列 A の固有値と固有ベクトルを求める
ginv(A)	行列 A の一般化逆行列を求める （ MASS パッケージの呼び出しが必要）
qr(A)	行列 A の QR 分解を行う
svd(A)	行列 A の特異値分解を行う
solve(A)	行列 A の逆行列を求める
t(A)	行列 A の転置を行う

行列計算

$$A = \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix}$$

```
> x <- c(1,2,3,4)
> A <- matrix(x, 2, 2)
> A * A                    # 対応する要素同士の積（≠行列の積）
     [,1] [,2]
[1,]    1    9
[2,]    4   16
> A %*% A                  # 行列の積
     [,1] [,2]
[1,]    7   15
[2,]   10   22
> t(A)                     # 行列 A の転置行列
     [,1] [,2]
[1,]    1    2
[2,]    3    4
```

行列へのアクセス

コマンド	機能
A[1,]	行列 A の 1 行目を取り出す
A[, 1]	行列 A の 1 列目を取り出す
A[1, 2]	行列 A の 1 行 2 列目の成分を 取り出す
A[c(1,2), 3]	行列 A の 1, 2 行 3 列目の成分を取り出す
A[c(1,2), c(2,3)]	行列 A の 1, 2 行目と 2, 3 列を取り出す

```
> A[1, ]
[1] 1 3 5
> A[1, 2]
[1] 3
```

演習問題①

1. 以下のような行列を作成し，変数 x に代入して下さい

```
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
```

2. 1. で作成した行列 x の 3 列目を抽出し，変数 y に代入して下さい

```
     [,1]
[1,]    3
[2,]    6
[3,]    9
```

本日のメニュー

1. 行列について
2. 複雑なくり返し文

複雑なくり返し文：例 1

- ある行列の各要素に対して「行番号+列番号」を代入することを考える
- まず，全ての要素が 0 である 2 × 3 の行列 x を作成する

```
> ( x <- matrix(rep(0,6), 2, 3) )

     [,1] [,2] [,3]
[1,]    0    0    0
[2,]    0    0    0
```

複雑なくり返し文：例 1

- 行列 x の各要素に「行番号+列番号」を代入する関数 myloop1() を定義することを考える → for 文の中でさらに for 文を使用する

```
> myloop1 <- function() {
+   x <- matrix(rep(0,6), 2, 3)
+   for (i in 1:2) {        # ループ①
+     for (j in 1:3) {      # ループ②
+       x[i,j] <- i + j
+     }
+   }
+   return(x)
+ }
> myloop1()
     [,1] [,2] [,3]
[1,]    2    3    4
[2,]    3    4    5
```

複雑なくり返し文：例 2

- 行列を引数として指定すると，「行番号+列番号」が偶数（ 2 で割った余りが 0 ）となる要素を 0 に置き換える関数を作成する
- まず，全ての要素が 1 である 2 × 3 の行列 x を作成する

```
> ( x <- matrix(rep(1,6), 2, 3) )

     [,1] [,2] [,3]
[1,]    1    1    1
[2,]    1    1    1
```

複雑なくり返し文：例 2

- 関数 myloop2() では， 2 つの for 文の中で if 文を適用することで「行番号+列番号」が偶数となる要素を 0 に置き換えている

```
> myloop2 <- function(x) {
+   y <- x
+   for (i in 1:nrow(x)) {
+     for (j in 1:ncol(x)) {
+       if ((i+j)%%2 == 0) y[i,j] <- 0
+     }
+   }
+   return(y)
+ }
> myloop2(x)
     [,1] [,2] [,3]
[1,]    0    1    0
[2,]    1    0    1
```

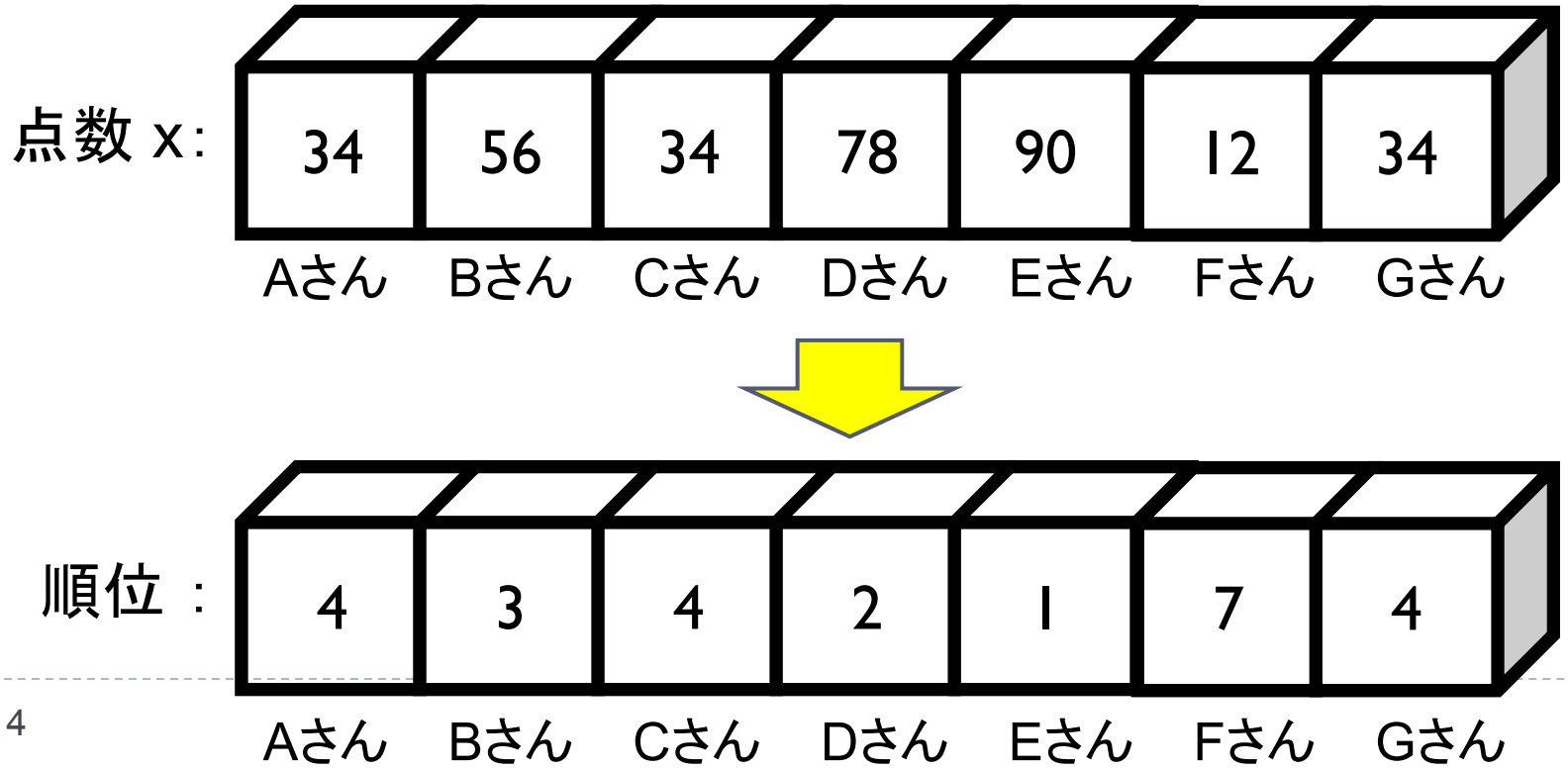
※ nrow(x)：x の行数、ncol(x)：x の列数

演習問題②

1. 行列を引数として指定すると，各要素に対して「行番号+列番号」をさらに加算する関数 myloop4() を定義して下さい

演習問題②

2. ある 7 人の生徒のテストの点数がベクトル x に格納されています
これから各生徒の順位を返す関数 myloop5() を定義して下さい
（同じ点数である場合，例えば 4 位に同じ点数の人が 3 人居る
場合は，3 人とも 4 位とし，次の順位は 7 位として下さい）



本日のメニュー

1. 行列について
2. 複雑なくり返し文

このスライドの内容が書籍になっています

- 舟尾 暢男 『 R で学ぶプログラミングの基礎の基礎（カットシステム）』

「R と RStudio のインストール」「R の基礎」「変数とベクトル」
「種々のベクトル」「関数の作成」「条件分岐」「くり返し」
「複雑なくり返し」「グラフ作成の概要」「問題集」「簡単なゲーム作成」

※ 本資料の演習問題の解答も載っています

このスライド以降の内容（続きは書籍にて）

- 問題集：イラスト付きで解説を載せています
  1. バブルソート
  2. 勝敗表
  3. あみだくじ
  4. 待ち行列
  5. 2 分探索
  6. 2 分法

このスライド以降の内容（続きは書籍にて）

- 簡単なゲームの作成
  1. ブラックジャック
  2. ○×ゲーム
  3. デイトレード

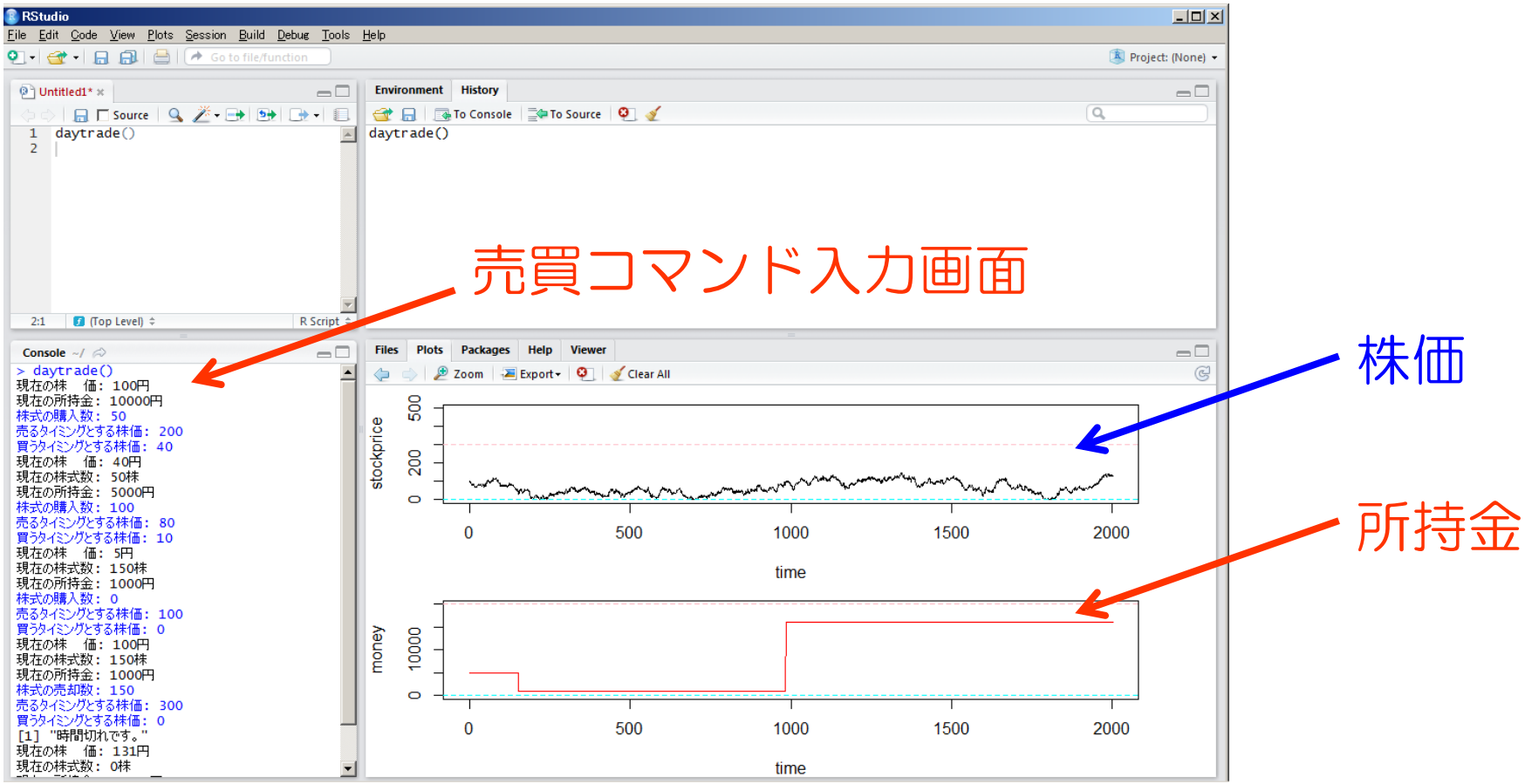
ブラックジャック

```
> bj()
手　: 8 11
点数: 18
もう1枚？（Y:Yes, N:No） n
手　: 8 11
点数: 18
相手の点数: 21
[1] "あなたの負けです。"
> bj()
手　: 7 11
点数: 17
もう1枚？（Y:Yes, N:No） N
手　: 7 11
点数: 17
相手の点数: バースト
[1] "あなたの勝ちです。
```

○×ゲーム

```
> marupeke()
     [,1] [,2] [,3]
[1,]    0    0    0
[2,]    0    0    0
[3,]    0    0    0
最初の手(行,列): 2,2
     [,1] [,2] [,3]
[1,]    2    0    0
[2,]    0    1    0
[3,]    0    0    0
次の手(行,列): 3,3
     [,1] [,2] [,3]
[1,]    2    2    0
[2,]    0    1    0
[3,]    0    0    1
次の手(行,列): ・・・
```

デイトレード



- 株の売り買いシミュレーションゲーム
- 所持金 10000 円を倍の 20000 円にすれば勝ち，一定時間が過ぎたら時間切れ

他にも楽しい演習問題が載っています

- 宝くじの当選番号を調べる関数の作成
- インターネット＋ケーブルテレビサービスの料金を計算する関数の作成
- 16 進数を 10 進数に変換する関数の作成
- サイコロを使ったシミュレーションを行う関数の作成
- ジャンケンを行う関数の作成
- 「ロト 6 」の番号が何等かを調べる関数の作成

等々

Rで学ぶプログラミングの基礎の基礎

終