



データ構造と型

Rには以下に列挙するようなデータ構造が用意されている

- スカラー
- ベクトル
- 行列(配列)
- リスト
- データフレーム

一つずつ例を見てみよう

1. スカラー

最も標準的なもの

```
> # 変数xに実数4を代入する
> x <- 4
> # 変数xの中身を表示する
> x
[1] 4
># [1]はデフォルトで表示の際についてくるだけなので無視
># 後にわかるが、[1]は要素の1つ目という意味で表示される
```

2. ベクトル(1)

ベクトルは関数 c() を用いて生成する

```
> # 変数xにベクトルを代入する
> x <- c(0,1,2,3,4)
> # 変数xの中身を表示する
> x
[1] 0 1 2 3 4
># ベクトルの2番目を表示したい場合
>x[2]
[1] 1
># 要素を参照するには[]を用いて要素番号を指定すればよい
```

2. ベクトル(2)

># 複数要素を指定する場合は番号をベクトルで指定する

```
>x[c(1,3,5)]
[1] 0 2 4
```

関数 c() の他にコロンを用いてベクトルを生成することができる

```
> # 変数xに1から8までの数字を代入する
> x <- 1:8
> # 変数xの中身を表示する
> x
[1] 1 2 3 4 5 6 7 8
```

3. 行列(1)

関数 matrix() を用いてベクトルを並べ替えて行列を生成する

```
> # 変数xに1から8までの数字が入ったベクトルを代入する
> x <- 1:8
> # 変数xの中身を表示する
> x
[1] 0 1 2 3 4 5 6 7 8
># 2行4列の行列に変換する(ベクトルの要素の順に注意)
> (A <- matrix(x,2,4))
      [,1] [,2] [,3] [,4]
[1,]  1    3    5    7
[2,]  2    4    6    8
```

括弧に意味があります

3.行列(2)

ベクトルの要素の順序を変えるには byrow オプションを用いる
(前スライドのAと比べるとわかりやすい)

```
>(B<-matrix(x,2,4,byrow=TRUE))
      [,1] [,2] [,3] [,4]
[1,]   1   2   3   4
[2,]   5   6   7   8

># 行列の和を計算する
>A + B
      [,1] [,2] [,3] [,4]
[1,]   2   5   8  11
[2,]   7  10  13  16
```

4.リスト(1)

異なる構造のデータをまとめて1つのオブジェクトとして扱えるようなもの
関数 list() を用いてリストを生成する

```
> x <- list(month=c(1,2,3),price=c(900,1000,1200),
            brand="DesignersProject")
> # 変数xの中身を表示する
> x
$month
[1] 1 2 3

$price
[1] 900 1000 1200

$brand
[1] "DesignersProject"
```

4.リスト(2)

```
> x$month # monthの内容を取り出す
[1] 1 2 3

> x$price[2] # priceの2番目の要素を取り出す
[1] 1000

> x[[3]] # xの3番目の項目(brand)を取り出す
[1] "DesignersProject"
```

4.リスト(3)

リストの各要素の名前は names を使ってあとから指定することができる

```
> y <- list(c(1,2,3),c(900,1000,1200),"DesignersProject")
> y
[[1]] # 要素名がついていないことがわかる
[1] 1 2 3

[[2]]
[1] 900 1000 1200

[[3]]
[1] "DesignersProject"
```

4.リスト(4)

```
> names(y) <- c("month","price","brand")
> # yの中身を表示
> y

$month # 各要素名がついている
[1] 1 2 3

$price
[1] 900 1000 1200

$brand
[1] "DesignersProject"
```

4.リスト(5)

```
> # names(変数) で変数名を表示できる
> names(y)
[1] "month","price","brand"
> names(y)[3] <- "company" # 3番目の項目の名前を変更する
> names(y)
[1] "month","price","company"
> y$price <- NULL # リストの要素を取り除く
> # yの中身を表示
> y # price要素が消えている
$month
[1] 1 2 3

$company
[1] "DesignersProject"
```

5. データフレーム(1)

データフレームはリストの一種
リストに含まれる値がすべて同じ長さのベクトルである特別なデータ構造
基本的には関数 `data.frame()` を用いてデータフレームを作成する

```
> # 各項目が同じ長さのベクトルからデータフレームを作る
> x <- data.frame( month = c(1,2,3), price =
+ c(900,1000,1200), deal = c(100,80,50))
> # 変数 x の中身を表示する
> x
  month price deal
1     1   900  100
2     2  1000   80
3     3  1200   50
```

5. データフレーム(2)

```
> x$price # price の項目を取り出す
[1] 900 1000 1200
> x[[2]] # リストの2番目の項目(ベクトル)を取り出す
[1] 900 1000 1200
> x[2] # 2列目だけからなるデータフレームを取り出す
  price
1   900
2  1000
3  1200
> x[2,] # 2行目(2月のデータ)を表示する
  month price deal
2     2   1000  80
```

補足

```
> x <- data.frame( month = c(1,2,3), price = c(900,1000,1200),
+ deal = c(100,80,50))
> x[[2]]
[1] 900 1000 1200
> x[2]
  price
1   900
2  1000
3  1200
> x[,2]
[1] 900 1000 1200
> x[2,]
  month price deal
2     2  1000  80
> class(x[,2])
[1] "numeric"
> class(x[2,])
[1] "data.frame"
> class(x[2])
[1] "data.frame"
```

5. データフレーム(3)

行列からデータフレームを作成する

```
> # 行列を作る
> x <- matrix( c(1,2,3,900,1000,1200,100,80,50), 3, 3 )
> x
      [,1] [,2] [,3]
[1,]    1   900   100
[2,]    2  1000    80
[3,]    3  1200    50
> # 関数 data.frame() を使って変換する
> z <- data.frame(x)
> z
  x1 x2 x3
1   1  900 100
2   2 1000  80
3   3 1200  50
> # データフレームには行名と列名が存在している 行名 1,2,3、列名 x1,x2,x3.
```

6. データ構造を調べる

オブジェクトのデータ構造を調べるには、関数 `is.vector()`, `is.matrix()`, `is.list()`, `is.data.frame()` などを使う

リストはベクトルとしても扱われるなど、1つのデータが複数の構造をもつ場合もあるので、注意が必要

```
> x <- 1
> is.vector(x) # スカラーもベクトルとして扱われる
[1] TRUE
> x <- c(1,2,3,4) # ベクトルを作成
> is.vector(x)
[1] TRUE
> is.matrix(x) # ベクトルは行列ではない
[1] FALSE
```