
Rでのデータ形式

Rのデータ形式

Rのデータ形式は非常に複雑なので、ここでは「データの型」と「データ構造」と分類しておく。

「データの型」

数値型(**numeric**), 文字型(**character**)など

これらを組み合わせて複雑なデータを定義できる。

「データ構造」 (データの集合)

ベクトル(**vector**), 行列(**matrix**), 配列(**array**), リスト(**list**),

データフレーム(**dataframe**)

Rのデータの型や構造を調べるコマンド: **typeof()**, **mode()**, **class()**

【実習1】 ①ベクトル

```
> vec1 <- c(1, 2, 3, 4)                                # オブジェクト名<-関数 (引数)
```

同様に, `vec2`として,

```
> vec2 <- c(5, 6, 7, 8)                                # オブジェクト名<-関数 (引数)
```

連続データ `c(1,2,3,4)` は `c(1:4)`, `c(5,6,7,8)` は `c(5:8)` でもいい。

単にデータ (ベクトル) を作るのであれば, `1:4`, `5:8` でもいい。

```
> vec1 + vec2                                           # ベクトルの和
```

```
[1] 6 8 10 12
```

```
> vec1 - vec2                                           # ベクトルの差
```

```
[1] -4 -4 -4 -4
```

```
> vec1*vec2                                             # 成分ごとの積のデータ
```

```
[1] 5 12 21 32
```

【実習1】 ①ベクトル

```
> vec1%*%vec2
```

```
[1,] 70
```

ベクトルの内積

```
> vec1[2];
```

```
[1] 2
```

vec1の第2要素

```
> vec1[1:3]
```

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

vec1の第1~第3要素の取出

```
> vec2[-2]
```

```
[1] 5 7 8
```

vec2の第2要素の削除

```
> length(vec1)
```

```
[1] 4
```

オブジェクトの長さ

【実習1】 ②規則的なデータ

```
> 1:5
```

```
[1] 1 2 3 4 5
```

```
> 5:1
```

```
[1] 5 4 3 2 1
```

```
> seq(1, 10, by=2)
```

```
[1] 1 3 5 7 9
```

```
> rep(c(1,2,3), times = 2)
```

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

```
> rep(c(1,2,3),each = 2)
```

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

```
> vec <- c(c(1, 2, 3), c(2, 3, 4))
```

```
> vec
```

```
[1] 1 2 3 2 3 4
```

複数のベクトルを結合したベクトルを作ることができる

【実習1】 ③切り上げ, 切り捨て

```
> x <- c(4.49, 4.5, 4.51, 5.5)
```

```
> ceiling(x)
```

```
[1] 5 5 5 6
```

```
> floor(x)
```

```
[1] 4 4 4 5
```

```
> round(x)
```

```
[1] 4 4 5 6
```

```
> floor(x + .5)
```

```
[1] 4 5 5 6
```

5.5はround関数の比較用

4.5が4に丸められていることに注意

四捨五入

【実習1】 ④乱数

> **runif(3)**

0と1の間の乱数3個

```
[1] 0.3263794 0.4356196 0.2098917
```

> **ceiling(runif(10)*6)**

1と6の間の自然数乱数10個

```
[1] 1 2 4 5 5 3 6 3 5 4
```

行列 (matrix)

【実習2】 行列コマンド

```
> m <- matrix(c(1 : 6), 2, 3)
```

行列のコマンド

```
> m
```

```
      [,1] [,2] [,3]  
[1,]  1   3   5  
[2,]  2   4   6
```

```
> length(m)
```

オブジェクトの長さ

```
[1] 6
```

```
> dim(m)
```

オブジェクトの次元

```
[1] 2 3
```

```
> nrow(m)
```

オブジェクトの行数

```
[1] 2
```

```
> ncol(m)
```

オブジェクトの列数

```
[1] 3
```

```
> class(m)
```

オブジェクトの型

```
[1] "matrix"
```

配列 (array)

> a <- array(1 : 4, dim = c(4)) # 1次元配列 (1行4列)

> a <- array(1: 6, dim = c(2, 3)) # 2次元配列 (2行3列)

> a <- array(1: 6,dim = c(2, 3, 2)) # 3次元配列

(各自で実行しよう)

【実習4】 (データフレーム)

```
> fr <- data.frame(name=c("青木","大木","加藤","山崎"), mark=c(20,24,12,32))
```

```
> fr # データフレーム作成
```

	name	mark
1	青木	20
2	大木	24
3	加藤	12
4	山崎	32

```
> mean(fr$mark) # 要素の操作：$でmark変数にアクセス (平均値)
```

```
[1] 22
```

【実習5】（リスト list）

> **list1** <- list("2年", "早稲田太郎", c(10, 20, 32)) # 異なる型にも対応

> **list1**

[[1]]

1 番目のリスト

[1] "2年"

[[2]]

2番目のリスト

[1] "早稲田太郎"

[[3]]

3番目のリスト

[1] 10 20 32

> **list1[[1]]**

1 番目を直接取り出すコマンド

[1] "2年"