

R のデータ構造

Andrew Ba Tran

Contents

ベクトル	1
データフレーム	2
リスト	3
オブジェクトを操作するための関数	4
演習	5

This is from the first chapter of learn.r-journalism.com.

ベクトル

ベクトルは、同じ基本型のデータ要素を順序付けた集まりです。ベクトルを構成する部分は、**コンポーネント**または**要素**と呼ばれます。

```
vec1 <- c(1,4,6,8,10)
```

```
vec1
```

```
## [1] 1 4 6 8 10
```

ベクトル`vec`は関数`c()`によって表されます。

```
vec1[5]
```

```
## [1] 10
```

ベクトルの要素は、`[i]`のようにインデックスを指定すれば表示できます。

```
vec1[3] <- 12 vec1
```

```
## [1] 1 4 12 8 10
```

このように、配列内の要素の一つが新しい数字に置き換えられています。

```
vec2 <- seq(from=0, to=1, by=0.25) vec2
```

```
## [1] 0.00 0.25 0.50 0.75 1.00
```

これは、ベクトルを作成するもう一つの方法である`seq()`、つまりシーケンス関数を示しています。

```
sum(vec1)
```

```
## [1] 35
```

データフレーム

スプレッドシートを使った作業に慣れている人は、R ではデータフレームが最も扱いやすいでしょう。

これは配列からデータフレームを作成する方法です。完全に理解する必要はありません。

スプレッドシートをインポートすると、データはあらかじめ構造化されています

```
patientID <- c(111, 208, 113, 408) age <-  
c(25, 34, 28, 52) sex <- c(1,2,1,1)  
diabetes <- c("Type1", "Type2", "Type1", "Type1") status <- c(1,2,3,1)  
patientdata <- data.frame(patientID, age, sex, diabetes, status) patientdata
```

```
##      patientID age sex diabetes status  
## 1      111 25   1     Type1     1 ## 2  
      208 34   2     Type2     2 ## 3  
      113 28   1     Type1     3 ## 4  
      408 52   1     Type1     1
```

これが実際に起こっていることです。一連の配列が作成され`data.frame()`という関数がデータフレーム構造に結合しているのです。

データフレームの要素を取得するには：

:は"から"という意味です。

```
patientdata[1:2]
```

```
##      patientID age  
## 1      111 25 ## 2  
      208 34 ## 3  
      113 28 ## 4  
      408 52
```

したがって1:2 は1 から2 という意味です。

```
patientdata[c("diabetes", "status")]
```

```
##      diabetes status  
## 1      Type1     1 ## 2  
      Type2     2 ## 3  
      Type1     3  
## 4      Type1     1
```

```
patientdata$age
```

```
## [1] 25 34 28 52
```

```
patientdata[1:2]
```

```
##      patientID age
## 1      111 25 ## 2
      208 34 ## 3
      113 28
## 4      408 52
```

```
patientdata[c(1,3),1:2]
```

```
##      patientID age
## 1      111 25 ## 3
      113 28
```

```
patientdata[2:3, 1:2]
```

```
##      patientID age
## 2      208 34
## 3      113 28
```

```
mean(patientdata$age)
```

```
## [1] 34.75
```

mean(patientdata[,2])の代わりに、patientdata\$age で列を参照することができます。

```
mean(patientdata$age)
```

```
## [1] 34.75
```

`patientdata`の`age`を参照するための別の方法にはこうしたものもあります。しかし、この方法はめったに使いません。

リスト

Rのもう一つの基本構造は *list* です。リストの最大の利点は、ベクトルの集まりである「列」（実際の順番として並んではない）が、行列やデータフレームとは異なり、同じ長さでなくてもよいことです。

JSON ファイルのようなものは構造化されています。

```
g <- "My First List"
```

```
h <- c(25, 26, 18, 39)
```

```
#次の行は、1 から(「:」)10 までの5 行の行列を作成しています。
```

```
j <- matrix(1:10, nrow = 5) k <- c("one", "two", "three") mylist <- list(title = g, ages = h, j, k)
```

こうすればリストを作業スペースに表示できます。

```
names(mylist)
```

```
## [1] "title" "ages" "" "" ""
```

リストにあるものを見つけるには

```
mylist[[2]]
```

```
## [1] 25 26 18 39
```

```
mylist[["ages"]][1]
```

```
## [1] 25
```

上記のコードはリストからデータを抽出します。

```
mylist$age + 10
```

```
## [1] 35 36 28 49
```

リストの番号を参照し使用するには、このようにします。

オブジェクトを操作するための関数

以下の `sample_df` データフレームでやってみましょう。

次のコードを実行してください。

```
sample_df <- data.frame(id=c(1001,1002,1003,1004), name=c("Steve", "Pam", "Jim",  
"Dwight"), age=c(26, 65), race=c("White", "Black", "White", "Hispanic"))  
sample_df$name <- as.character(sample_df$name)
```

```
sample_df
```

```
##      id  name age  race  
## 1 1001 Steve 26  White  
## 2 1002   Pam 65  Black  
## 3 1003   Jim 15  White  
## 4 1004 Dwight 7  Hispanic
```

`length(x)` - オブジェクトまたは配列に要素が何個あるかを調べます。

```
length(sample_df$name)
```

```
## [1] 4
```

`nchar(x)` - `x` が文字列の場合、何文字あるかを調べます。

```
sample_df$name[1]
```

```
## [1] "Steve"
```

```
nchar(sample_df$name[1])
```

```
## [1] 5
```

`dim(x)` - `x` の列数と行数を表示します。

```
dim(sample_df)
```

```
## [1] 4 4
```

`ncol(x)` - 列を数えます。

```
ncol(sample_df)
```

```
## [1] 4
```

nrow(x) - x の行数を返します。

```
nrow(sample_df)
```

```
## [1] 4
```

str(x) - x の構造を表示します。

```
str(sample_df)
```

```
## 'data.frame':      4 obs. of 4 variables:
## $ id : num 1001 1002 1003 1004
## $ name: chr "Steve" "Pam" "Jim" "Dwight"
## $ age : num 26 65 15 7
## $ race: Factor w/ 3 levels "Black","Hispanic",...: 3 1 3 2
```

summary(x) - R が理解しうる範囲でオブジェクトを要約します。

```
summary(sample_df)
```

```
##           id           name           age           race
## Min.      :1001   Length:4       Min.       : 7.00   Black      :1
## 1st Qu.:1002   Class :character   1st Qu.:13.00   Hispanic:1
## Median :1002   Mode  :character   Median :20.50   White     :2
## Mean      :1002                               Mean      :28.25
##
## 3rd Qu.:1003                               3rd Qu.:35.75
## Max.      :1004                               Max.      :65.00
```

View(x) - RStudio で閲覧するためにオブジェクトを開くコマンドです。

```
View(sample_df)
```

rm(x) - x を削除します。

```
rm(sample_df) sample_df
```

```
## Error in eval(expr, envir, enclos): object 'sample_df' not found
```

演習

[演習](<http://code.r-journalism.com/chapter-1/#section-data-structures-in-r>) を通じてこのセクションの知識を身につけましょう。

エクササイズアプリの実行に関する説明は、このセクションの[紹介ページ](http://learn.r-journalism.com/en/how_to_use_r/)にあります。