

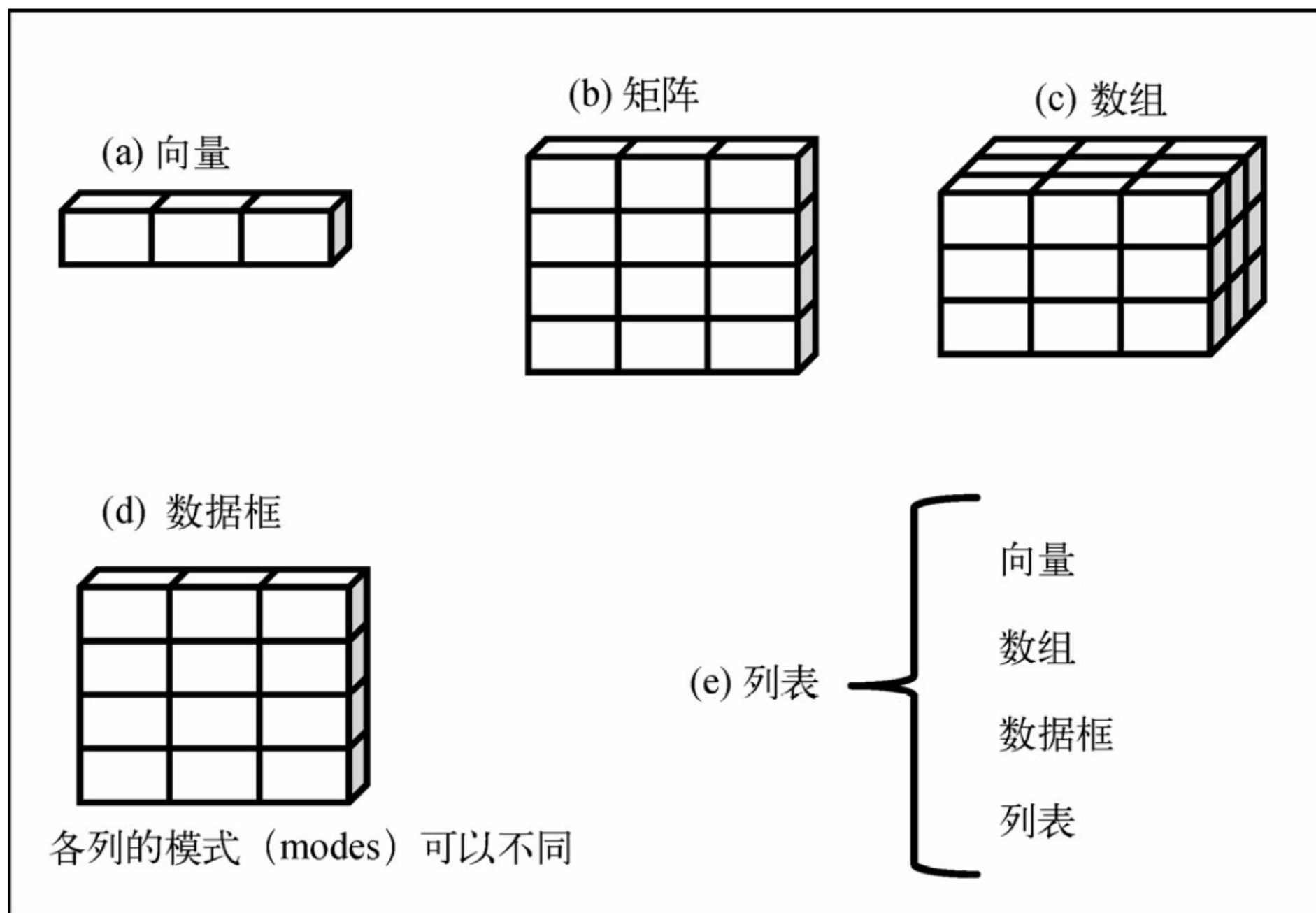
R语言数据对象I

- 课程简介:
- 数据分析简介: 数据; 数据分析; 过程; 目的; 数据展现; 数据分析报告;
- R软件简介: 历史; 优点; 安装; R Gui; RStudio;
- R基本使用: 解释性语言; >; <-; #, 表达式; 区分大小写;
- demo: *graphics; image; Hershey; persp;*
- 包函数: *install.package(); library(); search(); update.package();*
- 工作空间函数: *getwd(); setwd(); history(); savehistory(); loadhistory();*
save.image(); *c(); mean(); sd(); cor(); plot();*
- 输入输出函数: *source(); sink(); dev.off(); pdf(); png(); jpeg(); bmp();* *map();*
- 帮助函数: *help.start(); help(); ?; example(); data();* *rnorm(); density();*
- R软件操作: *DA01.R* *option(); summary(); hist(); runif();*

表2-1 病例数据

病人编号 (PatientID)	入院时间 (AdmDate)	年龄 (Age)	糖尿病类型 (Diabetes)	病情 (Status)
1	10/15/2009	25	Type1	Poor
2	11/01/2009	34	Type2	Improved
3	10/21/2009	28	Type1	Excellent
4	10/28/2009	52	Type1	Poor

- 数据集通常是由数据构成的一个矩形数组，行表示观测，列表示变量。
- R中有许多用于存储数据的结构，包括标量、向量、数组、数据框和列表。
- R可以处理的数据类型（模式）包括数值型、字符型、逻辑型（TRUE / FALSE）、复数型（虚数）和原生型（字节）。
- 在R 中，对象（object）是指可以赋值给变量的任何事物，包括常量、数据结构、函数，甚至图形。



- 定义
- 提取
- 操作



图2-1 R中的数据结构

- **向量**是用于存储数值型、字符型或逻辑型数据的一维数组。执行组合功能的函数**c()** 可用来创建向量。
- **标量**是只含一个元素的向量，例如**f<-3** 、 **g <- "US"** 和 **h <- TRUE** 。它们用于保存**常量**。
- 通过在**方括号**中给定元素所处**位置的数值**，我们可以访问向量中的元素。
- 使用**冒号(:)**用于生成一个数值序列。

- `a <- c(1, 2, 5, 3, 6, -2, 4)` 数值类型
- `b <- c("one", "two", "three")` 字符类型
- `c <- c(TRUE, FALSE, TRUE, FALSE)` 逻辑类型


定义

提取

- `a <- c(1, 2, 5, 3, 6, -2, 4)`
- `a[3]`
- `a[c(1, 3, 5)]` 
- `a[2:6]` 

`c()`可以将不同的向量合并成一个更长的向量

```
> y <- c(1,5,2)
> z <- c(x,0,y)
> z
[1] 8.2 3.7 4.5 5.6 7.3 0.0 1.0 5.0 2.0
```



: 可以生成步长为1的等差数列 (向量)

```
> 1:5
[1] 1 2 3 4 5
> 5:1
[1] 5 4 3 2 1
> 1.2:5
[1] 1.2 2.2 3.2 4.2
> 5.2:1
[1] 5.2 4.2 3.2 2.2 1.2
```

: 运算优先于四则运算

```
> 1:9-1
[1] 0 1 2 3 4 5 6 7 8
> 1:(9-1)
[1] 1 2 3 4 5 6 7 8
```

- **seq()**函数用来生成等距间隔的数列。
- 基本形式是: **seq(from=value1, to=value2, by=value3)**, 表示从value1开始, 到value2结束, 中间间隔为value3;

- 另一个使用形式为: **seq(length=value2, from=value1, by=value3)**。  默认设置

```
> seq(-2, 2, 0.5)
[1] -2.0 -1.5 -1.0 -0.5  0.0  0.5  1.0  1.5  2.0
> seq(length=9, from=-2, by=0.5)
[1] -2.0 -1.5 -1.0 -0.5  0.0  0.5  1.0  1.5  2.0
```

- **rep()**函数将一个向量重复若干次再放入新的变量。
- 使用形式为: **rep(x, times=n)**, 表示将x重复n次后构成的向量

```
> rep(2, 3)
[1] 2 2 2
> x <- 1:3
> rep(x, 3)
[1] 1 2 3 1 2 3 1 2 3
```



```
> x <- c(1,5,2,4,6,2,7,9,8,5)
```

```
> x[5]
```

```
[1] 6
```

```
> x[1:3]
```

```
[1] 1 5 2
```

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

```
[1] 1 2 6
```

```
> x[-5]
```

```
[1] 1 5 2 4 2 7 9 8 5
```



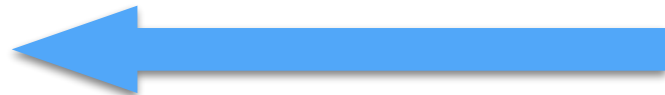
负(-)代表去除

+	加
-	减
*	乘
/	除
^, **	求幂
x %% Y	求余
x %/% Y	整除

```
> x <- c(1,3,5)
> y <- c(2,4,6)
> x+y
[1] 3 7 11
> x-y
[1] -1 -1 -1
> x*y
[1] 2 12 30
> x/y
[1] 0.5000000 0.7500000 0.8333333
> x^2
[1] 1 9 25
> x**2
[1] 1 9 25
> x%/%y
[1] 0 0 0
> x%%y
[1] 1 3 5
> x%**y
      [,1]
[1,] 44
```



向量运算的概念



<code>>,<</code>	大于，小于
<code><=, >=</code>	大于等于，小于等于
<code>!=, ==</code>	不等于，等于
<code>!x</code>	非x
<code>x y</code>	x或者y
<code>x & y</code>	x和y
<code>isTRUE(x)</code>	x是否为TRUE

```
> x <- 1:7
> x
[1] 1 2 3 4 5 6 7
> l <- x>3
> l
[1] FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE
> z <- c(TRUE, FALSE, F, T)
> all(c(1,2,3,4,5,6,7)>3)
[1] FALSE
> any(c(1,2,3,4,5,6,7)>3)
[1] TRUE
```

<i>length(object)</i>	显示对象中元素 / 成分的数量
<i>dim(object), str</i>	显示某个对象的维度 / 结构
<i>class(object)</i>	显示某个对象的类和类型
<i>name(object)</i>	显示某个对象中各成分的名字
<i>head(object), tail</i>	★ 列出某个对象的开始部分 / 最后部分
<i>cbind(object, object, ...), rbind</i>	按列 / 行合并对象
<i>object, ls(), rm(object,object,...), c(object,object,...)</i>	



- `length()`: 对象长度。
- `max()`: 向量中的最大值。
- `min()`: 向量中的最小值。
- `range()`: 边界
- `sum()`: 求和。
- `prod()`: 内积。

```
> x <- c(1,5,7,3)
> length(x)
[1] 4
> min(x)
[1] 1
> max(x)
[1] 7
> range(x)
[1] 1 7
> sum(x)
[1] 16
> prod(x)
[1] 105
```

- 向量排序函数包括：
`order()`, `sort()`, `sort.list()`等。
- `sort()`输出排序后的向量，`order()`和`sort.list()`返回下标排列。
- `which.max()`和
`which.min()`返回下标值。

```
> x <- c(10,6,4,7,8)
> order(x)
[1] 3 2 4 5 1
> sort(x)
[1] 4 6 7 8 10
> sort.list(x)
[1] 3 2 4 5 1
> which.min(x)
[1] 3
> which.max(x)
[1] 1
```


- 矩阵是一个二维数组，每个元素都拥有相同的模式（数值型、字符型或逻辑型）。可通过函数matrix 创建矩阵。

matrix()

```
myymatrix <- matrix(vector, nrow=number_of_rows, ncol=number_of_columns,  
                    byrow=logical_value, dimnames=list(  
                      char_vector_rownames, char_vector_colnames))
```

- 其中vector 包含了矩阵的元素，nrow 和ncol 用以指定行和列的维数，dimnames 包含了可选的以字符型向量表示的行名和列名。
- 选项byrow 则表明矩阵应当按行填充（byrow=TRUE）还是按列填充（byrow=FALSE），默认情况下按列填充。←
- 以使用下标和方括号来选择矩阵中的行、列或元素。X[i,] 指矩阵X 中的第i 行，X[,j]指第j 列，X[i,j] 指第i 行第j 个元素。选择多行或多列时，下标i和j可为数值型向量。

- `y <- matrix(1:20, nrow = 5, ncol = 4)`
 - `y`
-
- `cells <- c(1, 26, 24, 68)`
 - `rnames <- c("R1", "R2")`
 - `cnames <- c("C1", "C2")`
 - `mymatrix <- matrix(cells, nrow = 2, ncol = 2, byrow = TRUE, dimnames = list(rnames, cnames))`
 - `mymatrix`
-
- `mymatrix <- matrix(cells, nrow = 2, ncol = 2, byrow = FALSE, dimnames = list(rnames, cnames))`
 - `mymatrix`

- `x <- matrix(1:10, nrow = 2)`
- `x`

定义

-
- `x[2,]`
 - `x[, 2]`
 - `x[1, 4]`
 - `x[1, c(4, 5)]`

提取



- 数组（array）与矩阵类似，但是维度可以大于2。数组可通过array 函数创建。

```
myymatrix <- matrix(vector, nrow=number_of_rows, ncol=number_of_columns,  
                    byrow=logical_value, dimnames=list(  
                      char_vector_rownames, char_vector_colnames))
```

- 其中vector 包含了数组中的数据，dimensions 是一个数值型向量，给出了各个维度下标的最大值，而dimnames 是可选的、各维度名称标签的列表。

- `dim1 <- c("A1", "A2")`
- `dim2 <- c("B1", "B2", "B3")`
- `dim3 <- c("C1", "C2", "C3", "C4")`
- `z <- array(1:24, c(2, 3, 4), dimnames = list(dim1, dim2, dim3))`
- `z`


- 由于不同的列可以包含不同模式（数值型、字符型等）的数据，数据框的概念较矩阵来说更为一般。数据框将是你在R中最常处理的数据结构。
- 病例数据集包含了数值型和字符型数据。由于数据有多种模式，无法将此数据集放入一个矩阵。在这种情况下，使用数据框是最佳选择。
- 数据框可通过函数`data.frame()`创建

```
mydata <- data.frame(col1, col2, col3,...)
```
- 其中的列向量`col1, col2, col3,...` 可为任何类型（如字符型、数值型或逻辑型）。每一列的名称可由函数`names`指定。

- `patientID <- c(1, 2, 3, 4)`
- `age <- c(25, 34, 28, 52)`
- `diabetes <- c("Type1", "Type2", "Type1", "Type1")`
- `status <- c("Poor", "Improved", "Excellent", "Poor")`
- `patientdata <- data.frame(patientID, age, diabetes, status)`
- `patientdata`

定义

提取

- `patientdata[1:2]`
- `patientdata[c("diabetes", "status")]`
- `patientdata$age` 

```
attach(mtcars)
  summary(mpg)
  plot(mpg, disp)
  plot(mpg, wt)
detach(mtcars)
```

```
with(mtcars, {
  summary(mpg, disp, wt)
  plot(mpg, disp)
  plot(mpg, wt)
})
```

可以暂时不看
后边需要的时候再看

```
> with(mtcars, {
  nokeepstats <- summary(mpg)
  keepstats <<- summary(mpg)
})
> nokeepstats
Error: object 'nokeepstats' not found
> keepstats
```


- 变量可归结为名义型、有序型或连续型变量
 - ✱ 名义型变量是没有顺序之分的类别变量
 - ✱ 有序型变量表示一种顺序关系，而非数量关系
 - ✱ 连续型变量可以呈现为某个范围内的任意值，并同时表示了顺序和数量
- 类别（名义型）变量和有序类别（有序型）变量在R中称为因子（factor）
- 函数**factor()**以一个整数向量的形式存储类别值，整数的取值范围是[1...k]（其中k 是名义型变量中唯一值的个数）
- 要表示有序型变量，需要为函数factor()指定参数**ordered=TRUE**
- 对于字符型向量，因子的水平默认依字母顺序创建，你可以通过指定**levels**选项来覆盖默认排序

- `patientID <- c(1, 2, 3, 4)`
- `age <- c(25, 34, 28, 52)`
- `diabetes <- c("Type1", "Type2", "Type1", "Type1")`
- `status <- c("Poor", "Improved", "Excellent", "Poor")`
- `diabetes <- factor(diabetes)`
- `status <- factor(status, order = TRUE)`
- `patientdata <- data.frame(patientID, age, diabetes, status)`
- `str(patientdata)` 显示对象结构
- `summary(patientdata)` 显示对象统计概要

可以
暂时
不看

- 列表 (list) 是R的数据类型中最为复杂的一种。一般来说，列表就是一些对象（或成分，component）的有序集合。列表允许你整合若干（可能无关的）对象到单个对象名下
-

- `g <- "My First List"`
- `h <- c(25, 26, 18, 39)`
- `j <- matrix(1:10, nrow = 5)`
- `k <- c("one", "two", "three")`
- `mylist <- list(title = g, ages = h, j, k)`
- `mylist`
- `mylist[[2]]`

- 对象名称中的句点 (.) 没有特殊意义。但美元符号 (\$) 却有着和其他语言中的句点类似的含义，即指定一个对象中的某些部分。例如，A\$x是指数据框A中的变量x。
- 将一个值赋给某个向量、矩阵、数组或列表中一个不存在的元素时，R将自动扩展这个数据结构以容纳新值。

```
> x <- c(8, 6, 4)
> x[7] <- 10
> x
[1] 8 6 4 NA NA NA 10
```

- R中没有标量。标量以单元素向量的形式现。
- R中的下标不从0开始，而从1开始。在上述向量中，x[1]的值为8。
- 变量无法被声明。它们在首次被赋值时生成。

向量	<i>c(..., recursive=FALSE),</i>
矩阵	<i>matrix(vector, nrow, ncol, byrow=FALSE, dimnames)</i>
数组	<i>array(vector, dimensions, dimnames)</i>
数据框	<i>data.frame(col1, col2,col3,...), \$</i>
列表	<i>list(object1, object2,...)</i>
类型判断设置★	<i>is.numeric(), is.integer(), is.logical(), is.character()</i> <i>as.numeric(), as.integer(), as.logical(), as.character()</i>
<i>attach(), detach(),with(), factor(), rep(x,n), mode()</i> ★	

提问时间！

孙惠平

sunhp@ss.pku.edu.cn

谢谢！

孙惠平

sunhp@ss.pku.edu.cn