

Основы анализа больших данных

—
Лекция 3. Матрицы в R.



Класс данных: матрица (matrix)

Матрица – двумерный массив, состоящий из элементов одного типа данных

Матрицы в R:

- могут быть разной размерности (см. arrays)
- матриц как таковых в R, по сути, нет.

Матрица – это вектор, уложенный по столбцам.

Создание матриц в R: `matrix()`

Для создания матрицы в R можно использовать функцию `matrix()`.

```
matrix(data = NA, nrow=1, ncol=1, byrow = FALSE, dimnames = NULL)
```

Аргументы функции:

`data` – массив данных (вектор)

`nrow` – число строк

`ncol` – число столбцов

`byrow` – определяет как элементы вектора `data` заполняют матрицу – по строкам или по столбцам (по умолчанию)

`dimnames` – список из двух компонент, первая из которых названия строк, а вторая – названия столбцов (по умолчанию имена строк и столбцов не задаются)

Пример:

```
> data <- 1:4
> m <- matrix(data, nrow=2, ncol=2, byrow=TRUE)
> m
     [,1] [,2]
[1,]    1    2
[2,]    3    4
```

Создание матриц в R: атрибут dim

Можно получить доступ к размерности матрицы, воспользовавшись функцией `dim()`:

```
> m
      [,1] [,2]
[1,]    1    2
[2,]    3    4
> dim(m)
[1] 2 2
```

или функциями `nrow()` и `ncol()`:

```
> c(nrow(m), ncol(m))
[1] 2 2
```

«Фокус»:

```
> dim(m) <- NULL; m
[1] 1 3 2 4
```

```
> dim(m) <- c(2,2); m
```

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

или так:

```
> attr(m, "dim") <- c(2,2); m
      [,1] [,2]
[1,]    1    2
[2,]    3    4
```

Создание матриц в R: функции cbind() и rbind()

Функции `rbind()` и `cbind()` позволяют присоединять матрицы друг к другу, соответственно, по строкам и по столбцам.

`cbind(A,B)` создает матрицу из матриц (векторов), приписывая справа к A матрицу (вектор) B

`rbind(A,B)` создает матрицу, приписывая снизу к A матрицу (вектор) B

Примеры:

```
> rbind(c(1,2),c(3,4))
     [,1] [,2]
[1,]    1    2
[2,]    3    4
```

```
> cbind(c(1,2),c(3,4))
     [,1] [,2]
[1,]    1    3
[2,]    2    4
```

```
> t(cbind(c(1,2),c(3,4)))
     [,1] [,2]
[1,]    1    2
[2,]    3    4
```

Примечание: аргумент `...` позволяет передавать любое количество объектов

Операции над матрицами

```
> m1 <- matrix(1:4, nrow = 2)
> m2 <- matrix(c(1,2,2,3), nrow = 2)
> m1
      [,1] [,2]
[1,]    1    3
[2,]    2    4
> m2
      [,1] [,2]
[1,]    1    2
[2,]    2    3
```

Позлементное умножение матриц:

```
> m1*m2
      [,1] [,2]
[1,]    1    6
[2,]    4   12
```

Элементарные математические функции (sqrt(), log(), abs() и др.) также применяются поэлементно.

Матричное произведение:

```
> m1 %*% m2
      [,1] [,2]
[1,]    7   11
[2,]   10   16
```

Операции над матрицами: функция `outer()`

Функция `outer(X, Y, FUN="*", ...)` применяет заданную операцию `FUN` к каждой паре элементов массивов (векторов) `X` и `Y`. Размерность итоговой матрицы `c(dim(X), dim(Y))`.

По умолчанию `FUN="*"`, т.е. функция `outer()` (или оператор `%o%`) осуществляет внешнее произведение массивов `X` и `Y`.

Примеры:

```
> x <- 1:9; x
[1] 1 2 3 4 5 6 7 8 9
> x %o% x
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
[1,]    1    2    3    4    5    6    7    8    9
[2,]    2    4    6    8   10   12   14   16   18
[3,]    3    6    9   12   15   18   21   24   27
[4,]    4    8   12   16   20   24   28   32   36
[5,]    5   10   15   20   25   30   35   40   45
[6,]    6   12   18   24   30   36   42   48   54
[7,]    7   14   21   28   35   42   49   56   63
[8,]    8   16   24   32   40   48   56   64   72
[9,]    9   18   27   36   45   54   63   72   81
```

Операции над матрицами: функция outer() - продолжение

Примеры (продолжение):

```
> x <- 1:9; names(x) <- x
> y <- 2:8; names(y) <- paste(y, ":", sep = "")
> outer(y, x, `^`)
      1  2   3   4   5   6   7   8   9
2:  2  4   8  16  32  64 128 256 512
3:  3  9  27  81 243 729 2187 6561 19683
4:  4 16  64 256 1024 4096 16384 65536 262144
5:  5 25 125 625 3125 15625 78125 390625 1953125
6:  6 36 216 1296 7776 46656 279936 1679616 10077696
7:  7 49 343 2401 16807 117649 823543 5764801 40353607
8:  8 64 512 4096 32768 262144 2097152 16777216 134217728
```

Функция `outer()` применима также для массивов типа `character`:

```
> x <- c("AB", "CD")
> y <- c(1,2)
> outer(x,y,paste)
      [,1] [,2]
[1,] "AB 1" "AB 2"
[2,] "CD 1" "CD 2"
```


Операции над матрицами: элементы лин.алгебры

Решение СЛАУ

Для решения системы линейных алгебраических уравнений $Ax = b$ в R есть функция `solve(A, b, ...)`.

A - квадратная невырожденная матрица коэффициентов;
x – вектор неизвестных значений;
b - вектор свободных коэффициентов.

Эта функция решает уравнение вида $A \%*\% x = b$ относительно неизвестного x , где b может быть как вектором, так и матрицей.

Если вектор b не задан, то по умолчанию он является единичной матрицей и функция `solve()` возвращает обратную матрицу к A .

Пример:

```
> A = matrix(c(3,4,4,4), nrow = 2);A
      [,1] [,2]
[1,]    3    4
[2,]    4    4
> b=c(1,0)
> solve(A,b) -> x; x
[1] -1  1
```

Проверка:

```
> A \%*\% x
      [,1]
[1,]    1
[2,]    0
```

Функции `backsolve(A, b)`, `forwardsolve(B, b)`, `det(A)`, `colSums(X, na.rm)`, `rowSums(X, na.rm)`, `colMeans(X, na.rm)`, `rowMeans(X, na.rm)`, `eigen(X, symmetric, only.values = T)`, `lower.tri(X, diag=F)`, `upper.tri(X, diag=F)`

Индексирование матриц

$A[i, j]$ – ссылается на элемент i -й строки и j -го столбца матрицы A .

```
> m <- matrix(1:10, ncol = 5)
> m
      [,1] [,2] [,3] [,4] [,5]
[1,]     1     3     5     7     9
[2,]     2     4     6     8    10
> m[1,3]
[1] 5
> m[2,]
[1] 2 4 6 8 10
> m[,4]
[1] 7 8
> m[1,3:4]
[1] 5 7
> m[1,] <- 0; m
      [,1] [,2] [,3] [,4] [,5]
[1,]     0     0     0     0     0
[2,]     2     4     6     8    10
> m[, -5] <- 1:8; m
      [,1] [,2] [,3] [,4] [,5]
[1,]     1     3     5     7     0
[2,]     2     4     6     8    10
```

Схлопывание размерности:

```
> m[,5] -> n; n
[1] 0 10
> is.matrix(n)
[1] FALSE
> is.vector(n)
[1] TRUE

> m[,5, drop=F] -> n; n
      [,1]
[1,]     0
[2,]    10
> is.matrix(n)
[1] TRUE
```

Многомерные матрицы (arrays)

```
> m3 <- 1:8
> dim(m3) <- c(2,2,2)
> m3
, , 1
    [,1] [,2]
[1,]    1    3
[2,]    2    4

, , 2
    [,1] [,2]
[1,]    5    7
[2,]    6    8
> class(m3)
[1] "array"
```

Основной способ создания массивов:

```
array(data = NA, dim = length(data), dimnames = NULL)
```

Индексация:

```
> m3[1,2,2]
[1] 7
```

