



# Программирование в среде R

Шевцов Василий Викторович,  
директор ДИТ РУДН, [shevtsov-vv@rudn.ru](mailto:shevtsov-vv@rudn.ru)

# Операторы

|           |                             |
|-----------|-----------------------------|
| x=a       | Присваивания (справа)       |
| x<- a     | Присваивания (справа)       |
| a->x      | Присваивания (слева)        |
| x==a      | Сравнение. Равно            |
| x!=a      | Сравнение. Не равно         |
| x>a       | Сравнение. Больше чем       |
| x<a       | Сравнение. Меньше чем       |
| x>=a      | Сравнение. Больше или равно |
| x<=a      | Сравнение. Меньше или равно |
| !x        | Not x                       |
| x   y     | x OR y                      |
| x & y     | x AND y                     |
| isTRUE(x) | test if X is TRUE           |

# Операторы сравнения

```
a <- 10
```

```
a == 5
```

```
a != 5
```

```
a <= 10
```

```
a > 7
```

```
> a <- 10
```

```
> a == 5
```

```
[1] FALSE
```

```
> a != 5
```

```
[1] TRUE
```

```
> a <= 10
```

```
[1] TRUE
```

```
> a > 7
```

```
[1] TRUE
```

TRUE / FALSE

# Классы данных в R

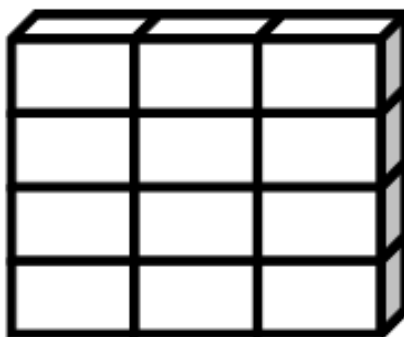
| Объект     | Возможные типы данных  | Использование в объекте нескольких типов данных |
|------------|--|---|
| vector     | числовой, символьный, комплексный, логический                              | Нет   |
| factor     | числовой, символьный   | Нет   |
| array      | числовой, символьный, комплексный, логический                              | Нет   |
| matrix     | числовой, символьный, комплексный, логический                              | Нет   |
| data.frame | числовой, символьный, комплексный, логичный                                | Да  |
| Ts         | числовой, символьный, комплексный, логический                              | Да  |
| List       | числовой, символьный, комплексный, логический, функция, выражение, формула | Да  |

# Классы данных в R

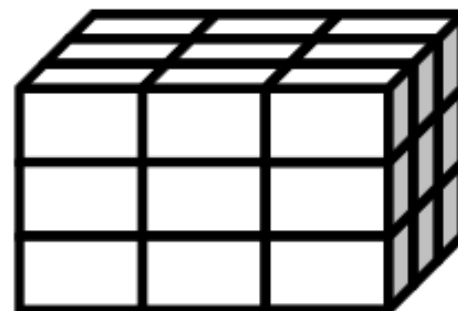
(a) Вектор



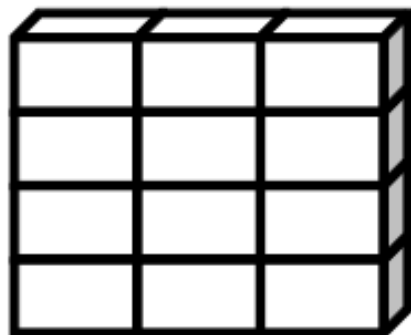
(b) Матрица



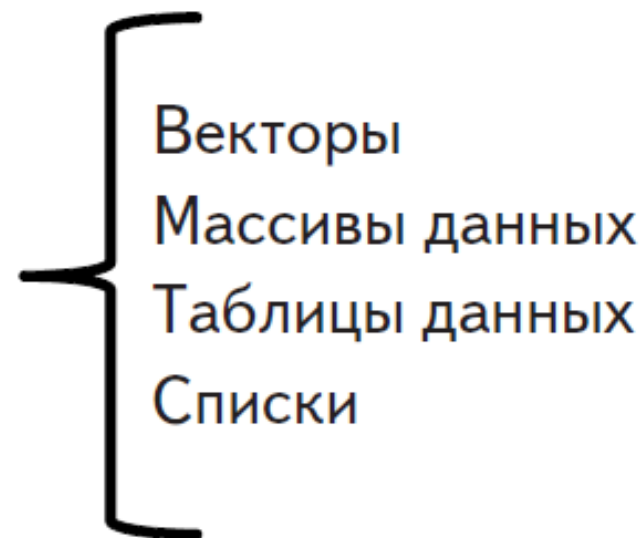
(c) Массив данных



(d) Таблица данных



(e) Список



Переменные могут иметь разный формат

# Векторы

## Создание вектора

```
> my_vector <- c(1,2,3,-5,-6)
```

```
> my_vector  
[1] 1 2 3 -5 -6
```

### Values

|           |                       |
|-----------|-----------------------|
| a         | 10                    |
| my_vector | num [1:5] 1 2 3 -5 -6 |

```
my_vector_2 <- c('a','b','c')  
my_vector_3 <- c(F,TRUE,FALSE)  
my_vector_4 <- c(1.23,3.45,10.43)
```

### Values

|             |                             |
|-------------|-----------------------------|
| a           | 10                          |
| my_vector   | num [1:5] 1 2 3 -5 -6       |
| my_vector_2 | chr [1:3] "a" "b" "c"       |
| my_vector_3 | logi [1:3] FALSE TRUE FALSE |
| my_vector_4 | num [1:3] 1.23 3.45 10.43   |



## Создание числовых последовательностей

- Если элементы числовой последовательности отличаются друг от друга на **единицу** (возрастающая последовательность), или на **-1** (убывающая последовательность), то можно задать только начальное а и конечное b значения искомой последовательности — a:b (разделяется двоеточием)

```
> x<-1:10;x  
[1] 1 2 3 4 5 6 7 8 9 10  
> x<-10:1;x  
[1] 10 9 8 7 6 5 4 3 2 1  
> y<-2:10;y  
[1] 2 3 4 5 6 7 8 9 10  
> y<-2.1:10;y  
[1] 2.1 3.1 4.1 5.1 6.1 7.1 8.1 9.1  
> y<--3.5:3.4;y  
[1] -3.5 -2.5 -1.5 -0.5 0.5 1.5 2.5  
> |
```



Обратите внимание на поведение построения последовательности при задании различных начальных и конечных значений



# Создание числовых последовательностей. rep()

- Функция rep() создает вектор с одинаковыми элементами:

```
> x<-rep(1,5);x  
[1] 1 1 1 1 1  
> |
```

- rep(a:b, times=n) создает вектор с повторяющимися последовательностями:

```
> x<-rep(2:5,times=2);x  
[1] 2 3 4 5 2 3 4 5  
> |
```

- rep(a:b, each=n) создает вектор с повторяющимися элементами:

```
> x<-rep(2:5,each=3);x  
[1] 2 2 2 3 3 3 4 4 4 5 5 5  
>  
> x<-rep(2:5,each=3,times=2);x  
[1] 2 2 2 3 3 3 4 4 4 5 5 5 2 2 2 3 3 3 4 4 4 5 5 5  
> |
```

## Создание числовых последовательностей. seq()

- `seq(from, to)` — аналогично `a:b`
- `seq(from, to, by= )` — задаются начальное, конечное значения и шаг последовательности
- `seq(from, to, length.out= )` — задаются начальное, конечное значения последовательности, а также общее число элементов последовательности. Шаг последовательности определяется самостоятельно  $by = ((to - from)/(length.out - 1))$
- `seq(x)` — последовательность строится от 1 до `x`.

```
> x<-seq(2,6);x
[1] 2 3 4 5 6
> x<-seq(2,6,by=0.5);x
[1] 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0
> x<-seq(2,6,length.out = 10);x
[1] 2.000000 2.444444 2.888889 3.333333 3.777778 4.222222 4.666667 5.111111 5.555556
[10] 6.000000
> x<-seq(10);x
[1] 1 2 3 4 5 6 7 8 9 10
> |
```

# Создание символьных последовательностей

- letters[from\_idx : to\_idx]
- LETTERS[from\_idx : to\_idx]

```
> letters[1:10]
[1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j"
> LETTERS[1:10]
[1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J"
> |
```

# Векторы. Адресация

|          |                          |
|----------|--------------------------|
| $x[n]$   | n-ый элемент             |
| $x[-n]$  | все элементы, кроме n-го |
| $x[1:n]$ | первые n элементов       |

```
> x<-2:11; x
[1] 2 3 4 5 6 7 8 9 10 11
> x[5]
[1] 6
> x[-5]
[1] 2 3 4 5 7 8 9 10 11
> x[1:4]
[1] 2 3 4 5
> |
```

# Векторы. Адресация

|                          |                                |
|--------------------------|--------------------------------|
| <code>x[-(1:n)]</code>   | все элементы, кроме первых n   |
| <code>x[c(1,4,2)]</code> | элементы с заданными индексами |
| <code>x["name"]</code>   | элемент с заданным именем      |

```
> x[-(1:4)]  
[1] 6 7 8 9 10 11  
> x[c(3,6,9)]  
[1] 4 7 10  
> |
```

```
> names(x)<-c("i1","i2","i3","i4","i5","i6","i7","i8","i9","i10")  
> x["i3"]  
i3  
4  
> |
```

```
> x<-1:10;x  
[1] 1 2 3 4 5 6 7 8 9 10  
> a<-length(x);a  
[1] 10  
> names(x)<-c(letters[1:a]);x  
a b c d e f g h i j  
1 2 3 4 5 6 7 8 9 10
```

# Векторы. Операторы сравнения

```
> x <- 1:10
> x
[1] 1 2 3 4 5 6 7 8 9 10
> x > 5
[1] FALSE FALSE FALSE FALSE FALSE TRUE TRUE TRUE TRUE TRUE
> x > 5 & x != 7
[1] FALSE FALSE FALSE FALSE FALSE TRUE FALSE TRUE TRUE TRUE
> |
```

## Векторы. Адресация

|   |  |
|---|--|
| <code>x[x &gt; 3]</code>                              | все элементы, большие 3                  |
| <code>x[x &gt; 3 &amp; x &lt; 5]</code>               | все элементы между 3 и 5                 |
| <code>x[x == 3]</code>                                | все элементы, равные 3                   |
| <code>x[x %in% c(2,3,4)]</code>                       | все элементы из заданного множества      |
| <code>x[c(logical_1, logical_2,...logical_n,)]</code> | с помощью logical вектора. $n=length(x)$ |

```
> x[x>4]
  i4  i5  i6  i7  i8  i9 i10
   5   6   7   8   9  10  11
```

```
> x[x>4&x<=10]
i4 i5 i6 i7 i8 i9
 5 6 7 8 9 10
```

```
> x
[1] 2 2 2 3 3 3 4 4 4 5 5 5 2 2 2 3 3 3 4 4 4 5 5 5
> x[x==3]
[1] 3 3 3 3 3 3
```

```
> x[x %in% c(2,3,4)]
[1] 2 2 2 3 3 3 4 4 4 2 2 2 3 3 3 4 4 4
```

```
> x<-1:10; x
[1] 1 2 3 4 5 6 7 8 9 10
> y<-c(TRUE,FALSE,FALSE,FALSE,FALSE,TRUE,TRUE,TRUE,TRUE,TRUE)
> x[y]
[1] 1 6 7 8 9 10
```

# Векторы. Операторы сравнения

```
> x <- 1:10
```

```
> x
```

```
[1] 1 2 3 4 5 6 7 8 9 10
```

```
> x > 5
```

```
[1] FALSE FALSE FALSE FALSE FALSE TRUE TRUE TRUE TRUE TRUE
```

```
> x > 5 & x != 7
```

```
[1] FALSE FALSE FALSE FALSE FALSE TRUE FALSE TRUE TRUE TRUE
```

```
>
```

|     |     |     |     |     |    |     |    |    |    |
|-----|-----|-----|-----|-----|----|-----|----|----|----|
| 1   | 2   | 3   | 4   | 5   | 6  | 7   | 8  | 9  | 10 |
| F   | F   | F   | F   | F   | T  | T   | T  | T  | T  |
| нет | нет | нет | нет | нет | да | да  | да | да | да |
| F   | F   | F   | F   | F   | T  | F   | T  | T  | T  |
| нет | нет | нет | нет | нет | да | нет | да | да | да |



# Арифметика векторов

```
> x <- 1:10
> x
 [1]  1  2  3  4  5  6  7  8  9 10
> x <- x + 10
> x
 [1] 11 12 13 14 15 16 17 18 19 20
```

```
> x <- x + 'a'
Error in x + "a" : non-numeric argument to binary operator
```

# Арифметика векторов

```
> x <- 1:10
> x
[1] 1 2 3 4 5 6 7 8 9 10
> x <- as.character(x)
> x
[1] "1" "2" "3" "4" "5" "6" "7" "8" "9" "10"
> x <- paste(x, 'a')
> x
[1] "1 a" "2 a" "3 a" "4 a" "5 a" "6 a" "7 a" "8 a" "9 a" "10 a"
```

```
> x <- 1:10
> x
[1] 1 2 3 4 5 6 7 8 9 10
> x <- paste(x, 'a')
> x
[1] "1 a" "2 a" "3 a" "4 a" "5 a" "6 a" "7 a" "8 a" "9 a" "10 a"
```

```
> x <- paste(x, 'a', sep='!')
> x
[1] "1!a" "2!a" "3!a" "4!a" "5!a" "6!a" "7!a" "8!a" "9!a" "10!a"
```

## Арифметика векторов

- Два вектора одинаковой длины могут быть добавлены, вычитаны, умножены или разделены, что дает результат в виде вектора вывода

```
> x<-c(3,5,7,9,2,4)
> y<-c(5,7,0,3,5,6)
> z<-x+y; z
[1] 8 12 7 12 7 10
```

```
> x<-c(3,5,7,9,2,4)
> y<-c(5,7,0,3,5,6)
> z<-y-x; z
[1] 2 2 -7 -6 3 2
```

```
> x<-c(3,5,7,9,2,4)
> y<-c(5,7,0,3,5,6)
> z<-y*x; z
[1] 15 35 0 27 10 24
```

```
> x<-c(3,5,7,9,2,4)
> y<-c(5,7,0,3,5,6)
> z<-y/x; z
[1] 1.6666667 1.4000000 0.0000000 0.3333333 2.5000000 1.5000000
```

# Арифметика векторов

```
> x <- 0:9
```

```
> y <- 10:15
```

```
> z <- x + y
```

Warning message:

In x + y : longer object length is not a multiple of shorter object length

```
> z
```

```
[1] 10 12 14 16 18 20 16 18 20 22
```

|    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|
| 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  |
| 10 | 11 | 12 | 13 | 14 | 15 | 10 | 11 | 12 | 13 |
| 10 | 12 | 14 | 16 | 18 | 20 | 16 | 18 | 20 | 22 |

## Арифметика векторов

- Если применить арифметические операции с двумя векторами разной длины, то элементы более короткого вектора переработаны для завершения операции.

```
> x<-c(3,5,7,9) [3,5]
```

```
> y<-c(5,7,0,3,5,6)
```

```
> z<-y+x;z
```

Warning message:

In y + x : longer object length is not a multiple of shorter object length

```
[1] 8 12 7 12 8 11
```

```
> z<-y-x;z
```

Warning message:

In y - x : longer object length is not a multiple of shorter object length

```
[1] 2 2 -7 -6 2 1
```

```
> z<-y*x;z
```

Warning message:

In y \* x : longer object length is not a multiple of shorter object length

```
[1] 15 35 0 27 15 30
```

```
> z<-y/x;z
```

Warning message:

In y/x : longer object length is not a multiple of shorter object length

```
[1] 1.6666667 1.4000000 0.0000000 0.3333333 1.6666667 1.2000000
```

# Манипуляция данными

- по индексу
- по имени
- по условию

```
> x<-1:10
> names(x)<-letters[1:10]
> x
  a  b  c  d  e  f  g  h  i  j
1  2  3  4  5  6  7  8  9 10
> x[3]<-300
> x[3]
  c
300
> x["c"]<-400
> x["c"]
  c
400
> x[3]
  c
400
```

```
> x[x>5]<-100
> x
  a  b  c  d  e  f  g  h  i  j
1  2 100  4  5 100 100 100 100 100
```

```
> x<-1:10
> names(x)<-letters[1:10]
> x[x>5]<-x[x>5]*100
> x
  a  b  c  d  e  f  g  h  i  j
1  2  3  4  5 600 700 800 900 1000
```

# Манипуляция вектором

- Изменение размерности вектора, вставка значений через оператор c()

```
> x<-1:10
> y<-c(x[1:5],100,x[6:10])
> y
 [1] 1 2 3 4 5 100 6 7 8 9 10
> z<-c(y,200,x);z
 [1] 1 2 3 4 5 100 6 7 8 9 10 200 1 2 3 4 5 6
[19] 7 8 9 10
> zz<-z[1:9];zz
 [1] 1 2 3 4 5 100 6 7 8
```

# Манипуляция вектором

## ■ Изменение размерности вектора

```
> x<-1:10
> names(x)<-letters[1:10];x
 a b c d e f g h i j
 1 2 3 4 5 6 7 8 9 10
> y<-c(x[1:5],100,x[6:10]);y
 a b c d e 100 f g h i j
 1 2 3 4 5 6 7 8 9 10
```

```
> x<-1:10
> names(x)<-letters[1:10]
> y<-100;names(y)<-"ee"
> z<-c(x[1:5],y,x[6:10]);z
 a b c d e ee f g h i j
 1 2 3 4 5 100 6 7 8 9 10
```

```
> z<-append(z,200);z
 a b c d e ee f g h i j
 1 2 3 4 5 100 6 7 8 9 10 200
```

```
> z<-append(z,300,after=3);z
 a b c 300 d e ee f g h i j
 1 2 3 4 5 100 6 7 8 9 10 200
```

```
> names(z)[4]<-"c1";z
 a b c c1 d e ee f g h i j
 1 2 3 300 4 5 100 6 7 8 9 10 200
> names(z)[13]<-"k";z
 a b c c1 d e ee f g h i j k
 1 2 3 300 4 5 100 6 7 8 9 10 200
```



## Ошибки

```
> x <- 1:10
```

```
> x[100]
```

```
[1] NA
```

```
> x[c(1:10,15,20)]
```

```
[1]  1  2  3  4  5  6  7  8  9 10 NA NA
```

# Матрицы

## Матрицы (matrix). 2-мерный массив

- Матрица в R — это специальный тип вектора, обладающий некоторыми добавочными свойствами (атрибутами), позволяющими интерпретировать его как совокупность строк и столбцов.
- Создание матрицы  
`m <- matrix(x, nrow = i, ncol = j)`

```
> x<-1:6;x  
[1] 1 2 3 4 5 6  
> ma<-matrix(x,nrow = 2,ncol = 3);ma  
      [,1] [,2] [,3]  
[1,]    1    3    5  
[2,]    2    4    6
```

```
> ma<-matrix(x, ncol=2, byrow=TRUE);ma  
      [,1] [,2]  
[1,]    1    2  
[2,]    3    4  
[3,]    5    6  
> ma<-matrix(x, ncol=2, byrow=FALSE);ma  
      [,1] [,2]  
[1,]    1    4  
[2,]    2    5  
[3,]    3    6
```

## Матрицы (matrix). 2-мерный массив

- Создание матрицы через атрибут dim

```
> x<-1:6;x
[1] 1 2 3 4 5 6
> attr(x,"dim")<-c(2,3);x
      [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
> x<-1:6;x
[1] 1 2 3 4 5 6
> attr(x,"dim")<-c(2,2);x
Error in attr(x, "dim") <- c(2, 2) :
  dims [product 4] do not match the length of object [6]
```

## Матрицы (matrix). 2-мерный массив

- Создание матрицы командами **cbind()**, **rbind()**
- Параметрами **cbind()** должны быть или векторы любой длины, или матрицы с одинаковым размером столбца, который является одинаковым числом строк.
- **rbind()** – аналогично, формируются строки

```
> ma<-cbind(0:5,5:10,10:15);ma
      [,1] [,2] [,3]
[1,]    0    5   10
[2,]    1    6   11
[3,]    2    7   12
[4,]    3    8   13
[5,]    4    9   14
[6,]    5   10   15
> ma<-cbind(0:5,5:9,10:13);ma
warning message:
In cbind(0:5, 5:9, 10:13) :
  number of rows of result is not a multiple of vector length (arg 2)
      [,1] [,2] [,3]
[1,]    0    5   10
[2,]    1    6   11
[3,]    2    7   12
[4,]    3    8   13
[5,]    4    9   10
[6,]    5    5   11
> x<-0:5;y<-5:10;z<-10:15
> ma<-cbind(x,y,z);ma
      x  y  z
[1,]  0  5 10
[2,]  1  6 11
[3,]  2  7 12
[4,]  3  8 13
[5,]  4  9 14
[6,]  5 10 15
> x<-0:5;y<-5:9;z<-10:13
> ma<-cbind(x,y,z);ma
warning message:
In cbind(x, y, z) :
  number of rows of result is not a multiple of vector length (arg 2)
      x y  z
[1,]  0 5 10
[2,]  1 6 11
[3,]  2 7 12
[4,]  3 8 13
[5,]  4 9 10
[6,]  5 5 11
```

## Матрица. Адресация

|  |  |
|--|--|
| <code>ma[n,m]</code>                       | Значение элемента матрицы                          |
| <code>ma[,m]</code>                        | Значение столбца                                   |
| <code>ma[n,]</code>                        | Значение строки                                    |
| <code>ma[1:n,2:m]</code>                   | Диапазон значений с... по.... для строк и столбцов |
| <code>ma[-(1:n),m]</code>                  | Кроме диапазона...                                 |
| <code>ma[c(1,2,4), c(1,2)]</code>          | Элементы с заданными индексами                     |
| <code>ma["RowName", "ColName"]</code>      | Значение элемента матрицы                          |
| <code>ma[ma &gt; 3]</code>                 | все элементы, большие 3                            |
| <code>ma[ma &gt; 3 &amp; ma &lt; 5]</code> | все элементы между 3 и 5                           |
| <code>ma[ma == 3]</code>                   | все элементы, равные 3                             |
| <code>ma[ma %in% c(2,3,4)]</code>          | все элементы из заданного множества                |

# Элементы матриц

```
> m1
      [,1] [,2] [,3]
[1,]    1    3    5
[2,]   20   100   60
> m1[1,2]
[1] 3
> m1[1,]
[1] 1 3 5
> m1[,2]
[1] 3 100
> m1[1,c(2,3)]
[1] 3 5
```

```
> dimnames(m1)<-list(c("row1","row2"),c("col1","col2","col3"))
> m1
      col1 col2 col3
row1    1    3    5
row2   20  100   60

> m1["row1","col2"]
[1] 3
> m1["row1",]
col1 col2 col3
   1    3    5
> m1[, "col2"]
row1 row2
   3  100
> m1["row1",c("col2","col3")]
col2 col3
   3    5
```

# Спасибо за внимание!



Шевцов Василий Викторович

shevtsov-vv@rudn.ru  
+7(903)144-53-57