Подмножества. Управляющие конструкции. Функции. Область видимости. Дата и время

TeXstudio Team

6 октября 2025 г.

Типы данных в R

Язык R работает со следующими типами данных:

* numeric – переменные, содержащие целочисленные значения (integer), действительные числа (double) и комплексные числа (complex);

```
double_value <- 290.7

double_value <- 290

integer_value <- 165L

complex_value <- 8 + 5i
```

Типы данных в R

* logical – переменные, содержащие логические значения: FALSE (сокращенно F) и TRUE (T);

```
is_monday <- F

bool_t <- TRUE
```

* character – текстовые переменные (отдельные значения таких переменных задаются в двойных либо одинарных кавычках);

```
text_value <- "Hello, Word"

char_value <- 'A'
```

Векторы

Векторы – это одномерный объект, которые могут хранить числовые, текстовые или логические значения (комбинации не допускаются)

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

b <- c("one", "two", "three", "four")

c <- c(TRUE, FALSE, TRUE, TRUE, FALSE)
```

Матрицы

Матрица – это двумерный массив данных, в котором все элементы имеют один и тот же тип (числовой, текстовый или логический). Матрицы создаются при помощи функции matrix. Общий синтаксис этой функции:

```
mymatrix <- matrix(
    вектор,
    nrow=число строк,
    ncol=число столбцов,
    byrow=логическое значение,
    dimnames=list(текстовый вектор с названиями строк,
    текстовый вектор с названиями столбцов)
)
```

Подмножества

Функция subset() – позволяет выбрать подбвыборку данных на основе какого-либо условия.
Вот два примера:

```
newdata <- subset(leadership, age >= 35 | age < 23, select=c(q1, q2, q3, q4))

newdata <- subset(leadership, gender=="M" \& age > 25, select=gender:q4)
```

Управляющие конструкции

Разновидности условных конструкций:

* if условный оператор if-else выполняет инструкцию, если заданное условие верно. Также есть возможность выполнить другую инструкцию, если условие не верно.
* ifelse - компактная и векторизованная версия оператора if-else.
* switch - оператор выбора switch выбирает инструкцию для выполнения в зависимости от значения выражения expr. Он имеет следующий синтаксис: switch(expr, ...) где многоточие (...) означает инструкции, соответствующие возможным значениям expr

Числовые и текстовые функции

Математические функции:

abs(x)	Модуль
sqrt(x)	Квадратный корень
ceiling(x)	Ближайшее целое число, не меньшее, чем x
floor(x)	Ближайшее целое число, не большее, чем x
trunk(x)	Целое число, полученное округлением x в сторону нуля
round(x, digits=n)	Округляет x до заданного числа знаков n после запятой
signif(x, digits=n)	Округляет x до заданного числа n значащих цифр

Числовые и текстовые функции

Статистические функции:

mean(x)	Среднее арифметическое
median(x)	Медиана
sd(x)	Стандартное отклонение
var(x)	Дисперсия
quantile(x, probs)	Квантили, где x – числовой вектор, для которого вычисляются квантили, а probs – числовой вектор с вероятностями в диапазоне [0; 1]
range(x)	Размах значений
sum(x)	Сумма

Пользовательские функции

```
y <- 10  # Глобальная переменная

my_function <- function(x) {
 result <- x + y  # y находится в глобальной области!
 return(result)
}

my_function(5)  # → 15
```

Область видимости

```
x <- "глобальная"

outer_func <- function() {
 x <- "внешняя"

 inner_func <- function() {
  x <- "локальная"
  print(x)  # Найдет "локальную"
 }
 inner_func()
}
```

Дата и время

```
as.Date(x, "input_format")
```

где x – это дата в текстовом формате, а input_format определяет формат представления даты

Циклы

Конструкция цикла for выполняет инструкцию statement для каждого значения в последовательности seq. Она имеет следующий синтаксис:

```
for (var in seq) {
    statement
}
```

Следующий код:

```
for (i in 1:5)  {
    print("Hello World")
}
```

выведет строку Hello World 5 раз.

Циклы

Конструкция цикла while повторно выполняет инструкцию, пока заданное условие остается истинным. Она имеет следующий синтаксис:

```
while (cond) {
 statement
}
```

Следующий код:

```
i <- 5
while (i > 0) {
 print("Hello World");
 i <- i - 1
}
```

тоже выведет строку Hello World 5 раз.

Название