Урок 6. Типы данных в R

В этом уроке мы обсудим типы данных в R. С двумя типами данных мы уже, на самом деле, познакомились, просто не останавливались подробно. Создавая переменные, мы работали с числовым (*numeric*) и строковым (*character*) типом. Вспомним:

```
age <- 32 # numeric
name <- "Ann" # character
```

Также в R существует логический тип. Переменная такого типа может принимать два значения `TRUE` и `FALSE`.

```
zero <- TRUE
one <- FALSE
```

Логический тип будет полезен для формулировки условий.

Обычно в программировании и анализе данных выделяют еще и целочисленный тип (*integer*). Целочисленный тип в R также существует, однако по умолчанию целые числа относятся к числовому типу. Числовой тип в R (*numeric*) – аналог *float* в Python, тоже хранит любые числовые значения.Чтобы сделать переменную целочисленной, нужно выполнить преобразование типов. Но прежде, давайте посмотрим, как узнать тип переменной. Для этого нам понадобится функция `class()`:

```
class(age)
```

```
## [1] "numeric"
```

```
class(name)
```

```
## [1] "character"
```

```
class(zero)
```

```
## [1] "logical"
```

Теперь самое время поговорить о преобразовании типов. Поскольку на практике мы редко работаем с переменными, в которых сохранено ровно одно число или одна строка, давайте рассмотрим его на примере векторов. Все функции для конвертации типов, которые мы рассмотрим далее, работают и для векторов, и для отдельных переменных.

Допустим, что мы выгрузили список чисел из файла и по каким-то причинам они считались некорректно, как строки. Приведем строковый тип к числовому:

```
nums <- c("2.5", "8", "9.1", "0")  # конечно, векторы могут быть не только числовыми
nums <- as.numeric(nums)
nums
```

```
## [1] 2.5 8.0 9.1 0.0
```

Все функции для преобразования типов устроены похоже: начинаются с префикса `as`, а далее следует название типа (*numeric*, *integer*, *character*, *logical*).

Разумеется, преобразование выше сработает только в том случае, если в качестве десятичного разделителя используется точка. В противном случае мы получим пропущенные значения (`NA`) и предупреждение:

```
nums2 <- c("2,5", "8", "9,1", "0")
nums2 <- as.numeric(nums2)
```

```
## Warning: в результате преобразования созданы NA
```

```
nums2
```

```
## [1] NA  8 NA  0
```

Если проблема только в разделителе, ее легко решить с помощью замены:

```
nums2 <- c("2,5", "8", "9,1", "0")
nums2 <- as.numeric(gsub(",", ".", nums2))
nums2
```

```
## [1] 2.5 8.0 9.1 0.0
```

Функция `gsub()` заменяет все вхождения одного символа или группы символов на другой. Применять ее можно как к переменным, так и к векторам и, как следствие, к столбцам в таблице. Сначала указываем, что заменяем, затем — на что и где.

Теперь рассмотрим другие примеры. Приведем числовой тип к целочисленному:

```
ints <- as.integer(c(2.0, 3.7, 3.0))
ints
```

```
## [1] 2 3 3
```

```
is.integer(ints)
```

```
## [1] TRUE
```

Обратите внимание, при превращении дробного числа в целое, дробная часть отбрасывается, округления не происходит (так же как и в Python).

Теперь приведем числовой тип к строковому:

```
ids0 <- 10:20 # пример числовой последовательности
ids0
```

```
##  [1] 10 11 12 13 14 15 16 17 18 19 20
```

```
ids <- as.character(ids0)
ids
```

```
##  [1] "10" "11" "12" "13" "14" "15" "16" "17" "18" "19" "20"
```

В завершение урока обсудим важные детали, связанные с векторами и типами данных. Как мы уже убедились, объекты одного типа можно объединять в векторы. Но вот сочетать типы в рамках одного векторы невозможно: один тип неизбежно «победит» другой. Например, строковый тип окажется сильнее числового и весь вектор станет строковым:

```
mixed1 <- c(2, "one", 6, 7)
mixed1
```

```
## [1] "2"   "one" "6"   "7"
```

```
class(mixed1) # да, функция class() работает для векторов тоже
```

```
## [1] "character"
```

```
is.character(mixed1)
```

```
## [1] TRUE
```

Или числовой тип окажется сильнее логического:

```
mixed2 <- c(1, 0, TRUE, FALSE)
mixed2
```

```
## [1] 1 0 1 0
```

```
class(mixed2)
```

```
## [1] "numeric"
```

Теперь пора приступить к практическому заданию. Желаю удачи!