Урок 2. Условные конструкции в R

В этом уроке мы посмотрим, как устроены в R условные конструкции. Сначала рассмотрим чисто технический пример с переменными, чтобы посмотреть на синтаксис, а затем перейдем к более интересным примерам с векторами.

Пусть в переменной `money` сохранено значение дохода в тысячах.

```
money <- 25
```

Напишем код, который позволит проверить, больше ли это число 30 или нет. Как известно из программирования на Python, условная конструкция начинается с оператора `if`, после которого следует условие, а затем в теле конструкции описываются действия, которые должны выполняться в случае, если условие верно. Однако условные конструкции в R сильно отличаются от таких конструкций в Python по синтаксису: само условие после `if` заключается в круглые скобки, а вместо отступов используются фигурные скобки.

```
if (money > 30){
print("Yes")
  }
```

Проверяемое условие записывается в круглых скобках после `if`, а тело конструкции помещается в фигурные скобки. Если в теле конструкции только одна строка кода, как здесь, фигурные скобки можно опустить. Если больше, этого делать нельзя, иначе R не поймет, что это продолжение условной конструкции.

Для сравнения – в Python код бы выглядел так:

```
money <- 25
if money > 30:
    print("Yes")
```

В этом примере код устроен таким образом, что в случае невыполнения условия ничего не происходит. Исправим это, добавив вторую, необязательную, часть условной конструкции с оператором `else`:

```
if (money > 30){
print("Yes")
  } else{
print("No")
}
```

```
## [1] "No"
```

Теперь, если условие неверно, на экран выводится слово “No”. Перечень действий после `else` тоже заключается в фигурные скобки, и если строка кода одна, ее можно опустить.

Для сравнения – в Python код бы выглядел так:

```
if money > 30:
    print("Yes")
else:
    print("No")
```

Разумеется, использовать сложные условия тоже можно:

```
if (money > 0 & money < 10000){
  print("Good")
}
```

```
## [1] "Good"
```

Примечание: оставлять закрывающую фигурную скобку на той же строке, что и код с выполняемой операцией, нормально, R не реагирует на положение скобок, но это не очень хорошо с точки зрения стиля кода.

Перейдем к векторам. Допустим, в векторе `cities` сохранен список городов. Наша задача: написать код, который в случае, если некоторый город в списке есть, выводит на экран сообщение "No need to add", а если нет, приписывает этот город в конец вектора.

Исходные данные:

```
cities <- c("Мурманск", "Москва", "Киров")
city <- "Омск"
```

Код:

```
if (city %in% cities){
  print("No need to add")
} else {
  cities <- c(cities, city)
}
cities
```

```
## [1] "Мурманск" "Москва"   "Киров"    "Омск"
```

Оператор `%in%` здесь — специальный оператор, который проверяет, принадлежит ли некоторый объект вектору (на самом деле, любой структуре данных, матрице или списку, например).

Функция `c()` уже встречалась нам в практическом задании прошлого модуля, она позволяет склеить два вектора или вектор с отдельным элементом, приписать его в конец (аналог метода `.append()` в Python).

На Python аналогичный код выглядел бы так:

```
if city in cities:
    print("No need to add")
else:
    cities.append(city)
cities
```

Здесь может возникнуть желание написать более универсальный код, чтобы проверять принадлежность города вектору не для отдельного города, а для нескольких городов из списка. И тут нам пригодятся циклы! Именно о них мы поговорим в следующем уроке, а пока выполните практическое задание по условным конструкциям.