

Практическое задание 5

Практическое задание не сдается на проверку и не оценивается.

Для выполнения задания вам потребуется познакомиться с двумя полезными функциями, одна из которых позволяет склеивать строки, а другая — создавать векторы из повторяющихся значений. Рассмотрим две задачи.

Задача 1. Для таблицы необходимо создать столбец с идентификаторами пользователей вида “user_01”. Всего пользователей 120, но их число может меняться.

Решение. Для удобства сохраним число пользователей в переменную, чтобы при его изменении не пришлось переписывать весь код.

```
n <- 120
```

Вспомним, как создать последовательность из целых чисел:

```
nums <- 1:n  
head(nums) # head - первые несколько элементов
```

```
## [1] 1 2 3 4 5 6
```

Для склеивания строк (или строк и чисел) понадобится функция `paste()`. Посмотрим на пример:

```
paste("new", "user", 1)
```

```
## [1] "new user 1"
```

В скобках перечисляются элементы, которые мы хотим склеить в одну строку. По умолчанию в качестве разделителя используется пробел. Если нам нужен другой разделитель, понадобится аргумент `sep`:

```
paste("new", "user", 1, sep="-")
```

```
## [1] "new-user-1"
```

Внутри функции `paste()` можно указывать не только отдельные строки или числа, но и векторы. «Приклеим» в начало каждого элемента вектора `nums` слово “user”:

```
users.tab <- paste("user", nums, sep="_")  
head(users.tab) # задача решена!
```

```
## [1] "user_1" "user_2" "user_3" "user_4" "user_5" "user_6"
```

Задача 2. Имеются 40 пользователей — участников эксперимента. Данные устроены так: сначала идут характеристики респондентов из группы А (их 20), затем — из группы В (их тоже 20). Необходимо создать столбец с идентификаторами такого вида: A1, A2, ..., A20, B1, B2, ..., B20.

Решение. Для решения этой задачи нам понадобится функция `rep()`. Она создает последовательности, повторяя элемент или вектор элементов нужное число раз. Например, так:

```
rep("R", 5)
```

```
## [1] "R" "R" "R" "R" "R"
```

Или так (дублируем вектор `c(1, 4, 5)`):

```
rep(c(1, 4, 5), 2)
```

```
## [1] 1 4 5 1 4 5
```

Если мы хотим повторить каждый элемент вектора несколько раз, например, получить подряд две 1, две 4 и две 5, нам понадобится аргумент `each`:

```
rep(c(1, 4, 5), each=2)
```

```
## [1] 1 1 4 4 5 5
```

Теперь применим полученные знания к решению этой задачи. Сначала создадим вектор с числами 1...20, 1...20:

```
nn <- rep(1:20, 2)
nn
```

```
## [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 1 2 3
## [24] 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
```

Теперь создадим последовательность из 20 букв А и 20 букв В:

```
ab <- rep(c("A", "B"), each = 20)
ab
```

```
## [1] "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A" "A"
## [18] "A" "A" "A" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B"
## [35] "B" "B" "B" "B" "B" "B"
```

Здесь опция `each = 20` означает, что мы хотим сначала 20 раз повторить первую букву в векторе, а затем 20 раз вторую.

Наконец, склеим полученные результаты:

```
paste(ab, nn, sep="")
```

```
## [1] "A1" "A2" "A3" "A4" "A5" "A6" "A7" "A8" "A9" "A10" "A11"
## [12] "A12" "A13" "A14" "A15" "A16" "A17" "A18" "A19" "A20" "B1" "B2"
## [23] "B3" "B4" "B5" "B6" "B7" "B8" "B9" "B10" "B11" "B12" "B13"
## [34] "B14" "B15" "B16" "B17" "B18" "B19" "B20"
```

1. Представьте, что вам для отчета необходимо получить следующую таблицу:

No	City	District
N1	Москва	Басманный
N2	Москва	Красносельский
N3	Москва	Басманный
N4	Москва	Красносельский
N5	Санкт-Петербург	Адмиралтейский
N6	Санкт-Петербург	Центральный
N7	Санкт-Петербург	Адмиралтейский
N8	Санкт-Петербург	Центральный

Создайте три вектора, которые могли бы послужить столбцами в такой таблице. Объедините их в датафрейм. Создавать вектора «вручную», просто перечисляя одинаковые элементы через запятую, нельзя: на практике обычно сталкиваемся с гораздо большим объемом данных и выполнять действия вручную нерационально.

Подсказка: для столбца *District* здесь пригодится конкатенация (склеивание) векторов. Приписать в конец одного вектора элементы другого можно с помощью уже известной функции `c()`:

```
# пример
a <- c(2, 3)
b <- c(0, 7)
```

```
c(a, b)
```

```
## [1] 2 3 0 7
```

2. Создайте (любым образом) матрицу, которая содержит число договоров, заключенных 3 компаниями за 6 месяцев:

```
##           [,1] [,2] [,3] [,4] [,5] [,6]
## company.1  223  100  350  122   99  346
## company.2  563  478  501  497  102  421
## company.3  156  187  122  134   53  190
```

2.1. Выведите общее число договоров, подписанных каждой компанией за все 6 месяцев.

2.2. Выведите общее число договоров, подписанных компаниями за каждый месяц.

2.3. Некоторые данные по компаниям оказались ошибочными. Измените некоторые элементы матрицы таким образом, чтобы она приобрела следующий вид:

```
##           [,1] [,2] [,3] [,4] [,5] [,6]
## company.1  223  110  350  122   99  346
## company.2  563  478  517  497  102  421
## company.3  156  187  127  134   53  190
```