**Лабораторная работа № 8 «Определение функций в R»**

Задание 1. Пусть функция decorate\_string действует поверх функции paste, дополнительно приклеивая к результату аргумент pattern. При этом этот аргумент должен быть присоединён как в начале строки (строк), так и в конце, но перевёрнутый задом наперёд. Определите функцию decorate\_string.

Примеры результатов вызовов функции

|  |  |
| --- | --- |
| **Вызов функции** | **Результат вызова** |
| decorate\_string(pattern = "123", "abc") | "123abc321" |
| decorate\_string(pattern = "123", "abc", "def") | "123abc def321" |
| decorate\_string(pattern = "123", c("abc", "def")) | "123abc321" "123def321" (вектор длины 2) |
| Обратите внимание, что функция decorate\_string должна помнить про аргументы для paste и правильно на них реагировать: | |
| decorate\_string(pattern = "123", "abc", "def", sep = "+") | "123abc+def321" |

Задание 2. Воспользуйтесь подготовленным шаблоном, чтобы получить две функции, содержащие честную и нечестную рулетку. Честная -- это когда все имеющиеся значения (всего их 37) выпадают с равной вероятностью. А нечестная пусть выдаёт все значения, кроме зеро, с равной вероятностью. Что же касается зеро (первый элемент определённого вектора roulette\_values), то вероятность его выпадения пусть будет в два раза больше, чем любого другого значения.

generator <- function(set, prob = rep(1/length(set), length(set))) {

}

roulette\_values <- c("Zero!", 1:36)

fair\_roulette <- generator(roulette\_values)

rigged\_roulette <- generator(<???>)

Функция generator – фабрика функций. Обратите внимание, как объявлен аргумент prob по умолчанию. В этом случае вызов generator(roulette\_values) без аргумента prob даст функцию для вызова честной рулетки fair\_roulette. Для нечестной рулетки rigged\_roulette вызов уточните самостоятельно. Рулетки проверяются вызовом обеих функций, так что убедитесь, что это функции, для которых обращение вида fair\_roulette(n) и rigged\_roulette(n) приведёт к получению n независимых бросков.

Задание 3. Требуется определить бинарный оператор. Пусть %+% действует на два числовых вектора, складывая их поэлементно, но без учёта правил переписывания: если длина векторов различна, то возвращаем вектор большей длины, но с пропущенными значениями в конце.

Например, 1:5 %+% 1:2 # c(2, 4, NA, NA, NA)

5 %+% c(2, 6) # c(7, NA)

Задание 4. В рамках исследовательского гранта сотрудники научно-исследовательского института Котологии занялись классификацией признаков сотен котов, котиков и котофеев. После упорного трёхлетнего труда они записали все найденные ими вариации признаков в следующие наборы:

cat\_temper <- c("задиристый", "игривый", "спокойный", "ленивый")  
cat\_color <- c("белый", "серый", "чёрный", "рыжий")  
cat\_age <- c("кот", "котёнок")  
cat\_trait <- c("с умными глазами", "с острыми когтями", "с длинными усами")

После этого они обратились к вам для составления всеобщего универсального каталога для включения его широко известный рецензируемый журнал "Вестник НИИК". Помогите учёным!

Составьте вектор cat\_catalogue, содержащий всевозможные комбинации имеющихся характеристик, и отсортируйте его. Помимо научной ценности, каталог можно использовать как средство для выбора домашнего питомца. Допустим, я хочу завести кота. Тогда я беру случайное число, скажем, 42, и выбираю запись под этим номером. Что за кот мне достался?