

Практическая работа №6

Создание пользовательских функций

В языке R достаточно просто создавать пользовательские функции, реализующие определенный алгоритм обработки данных, с целью многократного вычисления результатов по заданным параметрам, его анализа, в том числе путем представления в графической форме.

Напомним, функцией в программировании называется программный модуль, имеющий один или несколько входных формальных параметров, используемых для реализации алгоритма функции, и возвращающий в точку вызова функции единственное вычисленное значение. Во многих языках программирования, кроме функций, имеются также подпрограммы, которые могут возвращать через аппарат параметров несколько вычисленных данных.

В языке R функции в некоторой степени могут выполнять роль как традиционных функций, так и традиционных подпрограмм. Правда, в последнем случае возвращаемые значения указываются в теле функции структурной переменной (вектором, матрицей, списком, фреймом) с использованием оператора, вернее функции `return()`.

Функции в R имеют имя, по этому имени осуществляется их вызов. Параметры функции записываются в круглых скобках после имени функции и разделяются друг от друга запятой. Различают позиционные параметры и ключевые.

При вызове функции значения позиционных параметров указываются в той же последовательности, как они были определены при объявлении (создании) функции. Ключевые параметры следуют после позиционных, их порядок несущественный, так как они указываются в форме:

`имя_ключевого_параметра=значение`

Имена ключевых параметров при вызове функции могут не указываться, достаточно просто привести в качестве фактических параметров их значения, однако в этом случае следует строго соблюдать ту последовательность параметров, которая была установлена при создании функции.

Если при создании функции для некоторых ключевых параметров были установлены значения по умолчанию и эти значения при вызове функции устраивают пользователя, их можно не указывать.

Значения параметров функции могут быть заданы именами объектов, литералами (константами), выражениями или с использованием других функций.

При обращении к функциям в R значения нечисловых параметров, как ключевых, так и позиционных, часто записывают в двойных кавычках,

В R есть функции, не имеющие параметров. При обращении к таким функциям круглые скобки по-прежнему обязательны.

Возвратимся к информации о создании пользовательских функций.

Заголовок функции имеет вид:

```
имя_функции<- function (формальные_параметры)
```

После заголовка функции следует ее тело. Оно обрамляется фигурными скобками. Открывающаяся скобка {определяет начало тела функции, закрывающаяся скобка} – его конец.

В теле функции могут указываться выражения и операторы, а также комментарии.

Если тело функции содержит только одно вычисляемое выражение, фигурные скобки можно не указывать.

Если предполагается, что функция должна возвращать несколько вычисленных значений, следует в теле функции сформировать структурную переменную, элементами которой должны быть найденные в теле функции вычисленные значения (определенные, например, с помощью оператора присваивания, функций формирования вектора `c()`, матрицы `matrix()` и т.п.). Результат, возвращаемый функцией, в таком случае должен указываться перед окончанием тела функции оператором `return` (структурная_переменная).

Подводя итоги, представим две схемы вызова функций из программного скрипта – когда функция возвращает одно значение или несколько. Схема вызова пользовательских функций показана на рисунке 45.

При описании пользовательских функций можно определить значения параметров, задаваемые по умолчанию. Если все параметры функции заданы по умолчанию при ее создании, при обращении к такой функции фактические значения параметров в круглых скобках можно не указывать.



Рисунок 1 Схема вызова пользовательских функций с возвращением единственного (1) и нескольких значений (2)

Пример

Разработать пользовательскую функцию, вычисляющую выражение:

$$f = c - x_1^\alpha - x_2^\alpha \quad (10)$$

Привести пример вызова разработанной пользовательской функции.

Решение

Текст скрипта:

```

2 # Разработка пользовательских функций
3 f <- function(x1, x2, c, a){
4   c - x1^a - x2^a
5
6 }
7 m <- f(2, 1, 1.2, 2)
8 m
9
10 school <- function(a){
11   b = a + a * a
12   print(b)
13 }
14 school(2)
15 school(12)
16 school(22)

```

При описании пользовательских функций можно определить значения параметров, задаваемые по умолчанию. Если все параметры функции заданы по умолчанию при ее создании, при обращении к такой функции фактические значения параметров в круглых скобках можно не указывать.

Пример

Разработать пользовательскую функцию, вычисляющую выражение:

$$g = c - x^a - y^a \quad (11)$$

Привести пример вызова разработанной пользовательской функции.

Решение

Текст скрипта:

```
18 g<-function(x=0.5,y=0.5,c=1,a=2){
19   c-x^a-y^a
20
21 }
22 r<-g()
23 r
```

Как было сказано, функции могут возвращать несколько значений.

Пример

Разработать пользовательскую функцию, вычисляющую сумму, произведение и разность двух чисел.

Привести пример вызова разработанной пользовательской функции.

Решение

Текст скрипта:

```
25 poly<-function(a,b)
26 {
27   sum<-a+b
28   prod<-a*b
29   razn<-a-b
30   y<-c(sum,prod,razn)
31 }
32 g1<-poly(3,7)
33 g1[1]
34 g1[2]
35 g1[3]
```

Самостоятельная работа

Часть 1

1. Напишите функцию, которая принимает на вход базу данных, считает, сколько в базе данных переменных числового типа (numeric), сколько – факторного (factor) и сколько – текстового (character) и возвращает поименованный вектор, содержащий результаты подсчета. Если не знакомы с функциями в R (объект function), можно просто написать код, который выполняет действия, описанные выше.

Пример поименованного вектора: res <- c("a" = 4, "b" = 5, "c" = 6).

Дана база данных, с которой нужно поработать:

```
id <- 1:3
country <- c("Flatland", "Wonderland", "Sphereland")
craziness <- c(20, 15, 18)
region_type <- c("A", "B", "A")
author <- c("Abbot", "Carroll", "Burger")
size <- c(10, 100, 30)
m <- cbind(id, country, craziness, region_type, author, size)
df <- as.data.frame(m)
```

2. Напишите функцию, которая принимает на вход базу данных, выбирает из нее числовые переменные (столбцы типа numeric) и сохраняет их в новую базу данных. Если не знакомы с функциями в R (объект function), можно просто написать код, который выполняет действия, описанные выше.

3. Попробуйте выполнить задания 1-2 без использования apply-функций (если Вы их и не использовали, можете двигаться дальше).

4. Напишите функцию, которая принимает на вход вектор значений и, если вектор числовой, возвращает его медиану, если нет, выводит на экран сообщение "Vector is not numeric, cannot compute the median". Если не знакомы с функциями в R (объект function), можно просто написать код, который выполняет действия, описанные выше.

Часть 2

Некоторые методы обработки временных рядов

В данной работе мы рассмотрим некоторые методы обработки данных типа временного ряда и задачи, связанные с обработкой такого рода структур.

Для данной самостоятельной работы необходимо загрузить некоторые данные временного ряда цен акций компании с сайта finance.yahoo.com при помощи пакета “quantmod”. Это производится путём выполнения следующего скрипта:

```
if ("quantmod" %in% rownames(installed.packages()) == FALSE) {
  install.packages("quantmod") }
library(quantmod)

if ("stringr" %in% rownames(installed.packages()) == FALSE) {
  install.packages("stringr") }
library(stringr)

# Мы хотим загрузить акции с данными наименованиями в yahoo
```

```
downloadable_stocks <- c("ATVI", "^IXIC")

# Функция получения фреймов с данными
quantmod::getSymbols(Symbols = downloadable_stocks,
                      src = "yahoo",
                      from = as.Date.character("1900-01-01"))

# Функция get() позволяет получить содержимое объекта по его
названию-строке

# Мы можем и не знать названия акций в скрипте, но всё равно работать
с ними

# при пользовательском вводе названий
df <- data.frame(get(downloadable_stocks[1]))

# Применяем регулярное выражение для поиска и удаления ненужных
символов
downloadable_stocks <- stringr::str_remove(downloadable_stocks,
"[:punct:\\^]")

# Удалим полученные объекты
rm(list = downloadable_stocks)
```

Теперь у нас имеется фрейм данных df, который содержит 6 колонок, в каждой из которых содержится временной ряд, связанный с ценами акций компании. В силах выполняющего это задание поменять на своё усмотрение в скрипте переменную downloadable_stocks на любое другое значение наименования акций из существующих на finance.yahoo.com.

Функция Альтера-Джонса

В задачах анализа временных рядов нередко возникает ситуация, при которой необходимо из данных извлечь колебательную составляющую для того, чтобы оставить только трендовые движения показателя. Данная задача успешно решается многими методами очистки данных от шума (сглаживание), поиска периодической зависимости в данных (спектральное преобразование Фурье, автокорреляционная функция).

В данной работе для реализации в виде функции предлагается рассмотреть специальную функцию для поиска периодических компонент временного ряда – функцию *Альтера-Джонса*:

$$a_{\tau} = \frac{1}{n-\tau} * \sum_{i=1}^{n-\tau} |y_{i+\tau} - y_i|, \quad (12)$$

где n – длина исходного ряда, τ – область определения функции или смещения для входного ряда y , y – входной временной ряд для которого исследуется периодичность.

Функция Альтера-Джонса по своей природе показывает минимум в точке τ^* – смещения, определяемого как «почти-период». Собственно данное значение является характерным временем повторения очередного цикла похожих событий на временном ряду.

Периодичность данных, в которых присутствует трендовая компонента по факту сложно исследуема, поэтому для того, чтобы в исходных данных искать периодическую зависимость необходимо предварительно подготовить и изъять трендовую зависимость из исходных данных, сохраняя качественные особенности колебаний.

Данную задачу в рамках самостоятельной работы предлагается решить с помощью следующих функций:

средняя арифметическая пропорция:

$$y_t = \ln \left(\frac{x_{t-dt} + x_{t+dt}}{2x_t} \right), \quad (13)$$

средняя геометрическая пропорция:

$$y_t = \ln \left(\frac{x_{t-dt} * x_{t+dt}}{x_t^2} \right), \quad (14)$$

средняя гармоническая пропорция:

$$y_t = \ln \left(\frac{2 * x_{t-dt} * x_{t+dt}}{x_t * (x_{t-dt} + x_{t+dt})} \right), \quad (15)$$

где x – исходный вектор временного ряда с трендом, t – индекс массива или вектора, dt – пробный сдвиг по индексам (или времени), определяющий степень извлечения информации из данных. Чем больше показатель dt тем меньше будет элементов в векторе y , и тем более явно будут выражены большие колебания в исходных данных.

Данные функции изъятия тренда стоит применять перед использованием функции Альтера-Джонса только в случае наличия в данных устойчивой трендовой составляющей.

Задание 1.

1. Внимательно изучить скрипт, загружающий финансовые данные в среду R. Выполнить данный скрипт в вашем проекте самостоятельной работы

2. Реализовать функцию исключения тренда:

`out_of_trend(x, dt, method = c("Arifm", "Geom", "Garm"))`,

которая принимает на вход ряд чисел x , пробные смещения dt , метод решения данной задачи $method$. Функция на выходе должна возвращать также ряд

чисел, величина которого меньше на $2 * dt$ (подумайте почему), который получился в результате применения одного из трёх методов.

Все методы в функции должны быть реализованы, выбор метода осуществляется только при вызове функции. В стандартном случае или в любом другом случае вызывать среднюю арифметическую пропорцию.

Исключительными случаями для данного метода являются: ввод вектора длины меньше 3, ввод dt больше чем $\text{ceiling}(x/2)-1$, ввод нечислового вектора, ввод нечислового пробного смещения. Обработать при реализации эти случаи.

Перед применением функции к ряду с трендом поднять исходный ряд по оси ординат на величину самого минимального значения вектора плюс единица. Это нужно для корректной работы натурального логарифма.

3. Задать векторы $t = \text{seq}(0, 10, 0.1)$ и $x = 2 * t + 3 + \sin(2 * t)$.

Подсчитать среднее от вектора x . Применить функцию извлечения тренда к вектору x и записать его в переменную xn . Подсчитать среднее xn . Дать ответ, почему среднее xn находится около нуля.

4. Реализовать функцию Альтера-Джонса

$\text{Alter-Johns}(y)$,

где y – временной ряд (вектор чисел) без тренда. Подумайте, по определению функции Альтера-Джонса, каким является область определения для неё, т.е. в каком диапазоне могут лежать значения t .

5. Применить функцию Альтера-Джонса к вектору xn . Какое t отвечает *локальному* минимуму функции Альтера-Джонса?

6. Применить разработанные функции к одной из колонок таблицы данных `df`, загруженных ранее, и сохранить результат работы функции Альтера-Джонса по данному ряду с исключенным трендом в отдельном векторе. Подставьте данный вектор в функцию `plot()` и наблюдайте за её поведением. В какой точке наблюдается первый локальный минимум.

Внимание. Методы из пунктов 2 и 4 могут быть реализованы в отдельных файлах `*.R`, перед тестированием они должны быть последовательно вызваны единожды для того, чтобы они сохранились в окружении глобальных функций.

Часть 3

Итерационный метод

Данное задание снова относится к реализации метода простой итерации из двух прошлых самостоятельных работ. В данной работе будет необходимо реализовать функцию, обеспечивающую решение СЛАУ методом простой итерации, и обработать все её исключительные случаи в процессе реализации.

Функция, реализующая метод простой итерации должна выглядеть следующим образом:

$$\text{SIM}(A, u_0, f, n_iter = 10e5, \epsilon = 10e-7)$$

Исключительные случаи метода были разобраны ранее в работе №5.

Задание 1.

Реализовать функцию, осуществляющую решение СЛАУ методом простой итерации, соблюдая поставленные ранее условия.

Контрольные вопросы

1. Что понимается под функцией в программировании. В чем особенности функций в R
2. Какие параметры используются в функциях
3. В чем отличие позиционных параметров от ключевых параметров
4. Чем отличаются формальные и фактические параметры. Где они указываются
5. Каким образом могут быть заданы параметры функции
6. Каким образом функция в R может вернуть несколько значений
7. Как оформляется тело функции