Тема №1. Введение в программирование на R

Юрченков Иван Александрович

2022-02-25

Table of Contents

# Введение в язык программирования R

Язык R – прикладной язык программирования для статистической обработки данных. Данный инструмент предназначен для быстрой разработки прототипов моделей, научных вычислений, статистического анализа данных и численных экспериментов. Является свободно распространяемым языком с открытым исходным кодом.

Механизм работы языка R – интерпретация команд и кода, т.е. построчный ввод команд в программу-интерпретатор или построчная интерпретация программ в отличие от компиляции. Данная особенность порождает как положительные, так и отрицательные стороны работы с данными.

Плюсы:

1. Быстрое написание кода для получения желаемого результата.
2. Порождаемая интерпретацией максимальная абстракция языка на уровне представления данных.
3. Высокая совместимость с другими языками программирования.
4. Заранее реализованные эффективные алгоритмы в виде стандартных методов и функций.

Минусы:

1. Большее время выполнения тех же по сложности операций по сравнению с компилируемыми языками.
2. Большие накладные расходы на использование памяти при использовании стандартных структур данных.
3. Отсутствие возможности быстро создать исполняемый файл, работающий без установленного ПО интерпретатора языка.

Сообщество программистов на языке R вносит огромный вклад в развитие и конкурентноспособность инструмента посредством разработки сторонних библиотек для решения актуальных задач анализа данных и работы с данными.

# Введение в написание программ на языке R

Написание кода на языке R производится посредством написания однострочных команд (функций или методов). На каждую написанную команду на языке R интерпретатор реагирует определённым образом в тот же момент времени, выстраивая диалог с пользователем.

Например, открыв терминал R при помощи специальной команды или запустив консольный режим работы R вы сможете в дилоговом режиме записывать команды построчно с нажатием клавиши Enter при завершении набора команды. В таком случае ваша история введенных команд сохранится, и интерпретатор произведет все шаги, которые вы поставили на выполнение, но также данные команды не будут сохранены в отдельном файле и воспроизводимость такой работы весьма спорна.

Например, печать строки в консоль:

print("This is new string")

## [1] "This is new string"

или подсчет значений простых арифметических выражений:

print(5 + 7 / 2 \* 19 %% 4)

## [1] 15.5

все данные действия не будут сохранены для дальнейшего повторного выполнения или изменения, однако результат их работы все же виден будет.

Во избежание подобных ситуаций, консольному режиму работы R предпочитают набор команд в файлах скриптов (исполяемые файлы интерпретатора с расширением **.R**), либо построение последовательности анализа в ноутбуках (набор блоков кода в файлах с расширением **.Rmd** с возможностью параллельно вести документацию исследования).

## Арифметические операции R и простейшие функции

R в любом режиме работы поддерживает обычное построчное вычисление арифметических операций, результаты которых можно вывести в консоль или на экран монитора. В качестве базовых операций выделяют следующие:

print(5 + 6) # Сложение

## [1] 11

print(15 - 10.6) # Вычитание

## [1] 4.4

print(20 / 10) # Деление

## [1] 2

print(10 \* 7) # Умножение

## [1] 70

print(11.4 %% 6) # Остаток от деления нацело на число

## [1] 5.4

print(11.4 %/% 6) # Деление нацело на число

## [1] 1

print(9 ^ 2) # Возведение в степень

## [1] 81

print(9 \*\* 2) # Возведение в степень

## [1] 81

Таким образом, вы можете достаточно просто считать определенные составные и сложные выражения просто при помощи консольного ввода вычислений в R:

print(5 + 6 \*(2 + 3) / 1.5)

## [1] 25

## Простейшие функции языка

Среди простейших функций языка выделим специальные начальные функции для написания простых программ. Среди них:

* **print()** – вывод в консоль содержимого имени (переменной) в переопределенном для каждого класса объекта виде. Например вывод для матриц и векторов будет различен. Этот вывод переопределен на уровне объявления класса матриц и векторов соответственно.

Пример:

vector\_1 <- 1:9  
print(vector\_1)

## [1] 1 2 3 4 5 6 7 8 9

matrix\_1 <- matrix(1:9, nrow = 3, byrow = TRUE)  
print(matrix\_1)

## [,1] [,2] [,3]  
## [1,] 1 2 3  
## [2,] 4 5 6  
## [3,] 7 8 9

* **cat()** – вывод в консоль содержимого имени (переменной) без привзяки к структуре объекта.

Пример:

vector\_1 <- 1:9  
cat(vector\_1)

## 1 2 3 4 5 6 7 8 9

matrix\_1 <- matrix(1:9, nrow = 3, byrow = TRUE)  
cat(matrix\_1)

## 1 4 7 2 5 8 3 6 9

* **sin()** и другие тригонометрические функции (см. справку по тригонометрическим функциям с помощью **help(“sin”)**) – вычисление тригонометрическх функций относительно чисел.

Пример:

print(sin(3.143))

## [1] -0.001407346

print(cos(3.143))

## [1] -0.999999

print(tan(1))

## [1] 1.557408

print(atan(1))

## [1] 0.7853982

* **paste()** – конкатенация нескольких строк или строк с вектором.

Пример:

print(paste("Hello", "fellows"))

## [1] "Hello fellows"

cat(paste("Hello", "my", "friends", sep = "\n"))

## Hello  
## my  
## friends

## История команд и образ проекта

Инициация запуска R влечет за собой автоматическое ведение запасного файла, в котором хранится история выполненных построчных команд с расширением **.Rhistory** а также ведение файла с расширением **.RData** с образом среды, сохраненными переменными, объектами, графиками и т.д.

При выходе из R вас встретит уведомления отдельно о сохранении истории команд и о сохранении образа рабочей области. Сохранить вручную можно при помощи специальных функций из R:

file\_history = "history\_of\_commands.Rhistory"  
savehistory(file = file\_history)  
  
file\_image = "objects\_of\_project.RData"  
save(file = file\_image)

Загрузить записанную историю команд и переменные из соответствующих файлов можно с помощью команд:

loadHistory(file = file\_history)  
load(file = file\_image)

Применение данных команд в большинстве случаев желательно только в случае использовании RGui стандартного пакета R, а также консольного режима из терминала операционной системы.

При использовании специального ПО IDE RStudio данные действия выполняются на фоне с периодичностью по умолчанию.