

Компьютерный практикум по статистическому анализу данных

Лабораторная работа № 1. Julia. Установка и настройка. Основные
принципы

Демидова Екатерина Алексеевна

Содержание

1	Введение	4
2	Теоретическое введение	5
3	Выполнение лабораторной работы	6
4	Выводы	10
	Список литературы	11

Список иллюстраций

3.1	Примеры	6
3.2	Изучение докумнтации	7
3.3	Задание 1	8
3.4	Задания 2 и 3	9

1 Введение

Цель работы

Основная цель работы – подготовить рабочее пространство и инструментарий для работы с языком программирования Julia, на простейших примерах познакомиться с основами синтаксиса Julia.

Задачи

1. Установите под свою операционную систему Julia, Jupyter.
2. Используя Jupyter Lab, повторите примеры из раздела 1.3.3.
3. Выполните задания для самостоятельной работы:
 1. Изучите документацию по основным функциям Julia для чтения / записи / вывода информации на экран: `read()`, `readline()`, `readlines()`, `readdlm()`, `print()`, `println()`, `show()`, `write()`. Приведите свои примеры их использования, поясняя особенности их применения.
 2. Изучите документацию по функции `parse()`. Приведите свои примеры её использования, поясняя особенности её применения.
 3. Изучите синтаксис Julia для базовых математических операций с разным типом переменных: сложение, вычитание, умножение, деление, возведение в степень, извлечение корня, сравнение, логические операции. Приведите свои примеры с пояснениями по особенностям их применения.
 4. Приведите несколько своих примеров с пояснениями с операциями над матрицами и векторами: сложение, вычитание, скалярное произведение, транспонирование, умножение на скаляр.

2 Теоретическое введение

Julia — высокоуровневый свободный язык программирования с динамической типизацией, созданный для математических вычислений.[1]. Эффективен также и для написания программ общего назначения. Синтаксис языка схож с синтаксисом других математических языков, однако имеет некоторые существенные отличия.

Для выполнения заданий была использована официальная документация Julia[2].

3 Выполнение лабораторной работы

Выполним примеры из лабораторной работы(рис. 3.1)

```
+
- for T in [Int8,Int16,Int32,Int64,Int128,UInt8,UInt16,UInt32,UInt64,UInt128]
-   println("${Lpad(T,7)}: [$(typemin(T)),$(typemax(T))]" )
- end

Int8: [-128,127]
Int16: [-32768,32767]
Int32: [-2147483648,2147483647]
Int64: [-9223372036854775808,9223372036854775807]
Int128: [-170141183460469231731687303715884105728,170141183460469231731687303715884105727]
UInt8: [0,255]
UInt16: [0,65535]
UInt32: [0,4294967295]
UInt64: [0,18446744073709551615]
UInt128: [0,340282366920938463463374607431768211455]

52.7 ms

16
- begin
-   function g(x)
-       x^2
-   end
-   g(4)
-   gg(x)=x^2
-   gg(4)
- end

9.8 ms

- a = [4 7 6] # вектор-строка
- b = [1, 2, 3] # вектор-столбец
- a[2], b[2] # вторые элементы векторов a и b

87.0 μs

1x1 Matrix{Int64}:
27
- begin
-   aa = 1; bb = 2; cc = 3; dd = 4 # присвоение значений
-   AAm = [aa bb; cc dd] # матрица 2 x 2
-   AAm[1,1], AAm[1,2], AAm[2,1], AAm[2,2] # элементы матрицы
-   aaa = [1 2]
-   AA = [1 2; 3 4]
-   aaa*AA*aaa'
- end
```

Рис. 3.1: Примеры

Для изучения документации достаточно поставить ? перед именем нужной функции(рис. 3.2).



Рис. 3.2: Изучение документации

При использовании основных функций Julia для чтения / записи / вывода информации(рис. 3.3) можно увидеть следующие особенности:

- `read()` – чтение одного элемента,
- `readline()` – чтение строки,
- `readlines()` – чтение всех строк файла и запись их в массив,
- `readdlm()` – считывает данные как матрицу с обязательно одинаковым количеством элементов в каждой строке(дописывает "", если элемента нет),
- `print()` – стандартный вывод без перевода строки,
- `println()` – стандартный вывод с переводом строки,
- `show()` – показывает внутренне представление данных(например, строковой тип с кавычками),
- `write()` – стандартная запись в файл.


```
begin
  an = 2
  bn = 11

  println(bn/an, " ", an\bn, " ", bn//an)
  println(bn//an+1//2)
  println(1/(0.0), " ", 1/(-0.0), " ", 0.0/0.0)
  println(an == bn)
  println(false+false-true-true)
  println(false|true, " ", false&true, "\n")

  Am = [1 2; 3 4]
  Bm = [1 1; 2 2]
  C = [2;2]

  println(Am*Bm)
  println(transpose(C)*Bm)
  println(Bm/Am)
  println(Am*bn)
  println(Bm^2, "\n", sqrt(Am))

end
```

```
5.5 5.5 11//2
6//1
Inf -Inf NaN
false
-2
true false

[5 5; 11 11]
[6 6]
[-0.5 0.5; -1.0 1.0]
[11 22; 33 44]
[3 3; 6 6]
ComplexF64[0.5536885671459113 + 0.4643941628390708im 0.8069607270132165 - 0.212
42647876640208im; 1.2104410905198246 - 0.31863971814960307im 1.764129657665736
+ 0.1457544446894677im]
```

3.4 ms

Рис. 3.4: Задания 2 и 3

4 Выводы

В результате выполнения работы подготовили рабочее пространство и инструментарий для работы с языком программирования Julia, на простейших примерах познакомились с основами синтаксиса Julia.

Список литературы

1. JuliaLang [Электронный ресурс]. 2024 JuliaLang.org contributors. URL: <https://julialang.org/> (дата обращения: 11.10.2024).
2. Julia 1.11 Documentation [Электронный ресурс]. 2024 JuliaLang.org contributors. URL: <https://docs.julialang.org/en/v1/> (дата обращения: 11.10.2024).