

Отчёт по лабораторной работе №1

Компьютерный практикум по статистическому анализу

Надежда Александровна Рогожина

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	14
	Список литературы	15

Список иллюстраций

4.1	whoami	8
4.2	read + примеры	9
4.3	write + примеры	9
4.4	readline + readlines + примеры	9
4.5	readdlm + примеры	10
4.6	print + println + примеры	10
4.7	show + примеры	11
4.8	parce	11
4.9	Основные математические операции	12
4.10	Создание матриц, умножение на скаляр	12
4.11	Операции с матрицами	13

Список таблиц

1 Цель работы

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

2 Задание

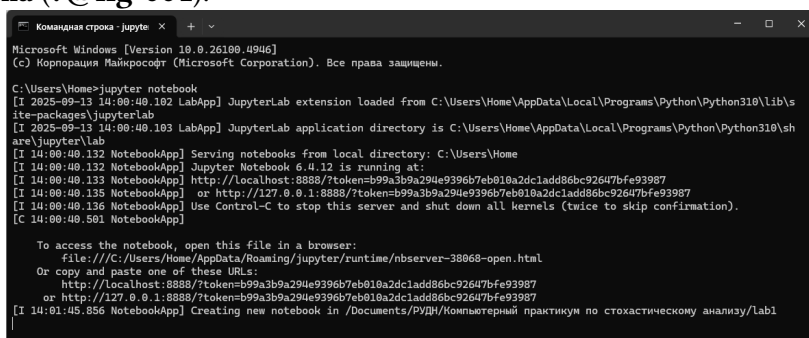
1. Установите под свою операционную систему Julia, Jupyter (разделы 1.3.1 и 1.3.2).
2. Используя Jupyter Lab, повторите примеры из раздела 1.3.3.
3. Выполните задания для самостоятельной работы (раздел 1.3.4).

3 Теоретическое введение

Научные вычисления традиционно требуют высочайшей производительности, однако эксперты по доменам в значительной степени перенесены на более медленные динамические языки для ежедневной работы. К счастью, методы современного языка и компилятора позволяют в основном устранить компромисс производительности и обеспечивать достаточно продуктивную среду для прототипирования и достаточно эффективно для развертывания применений, интенсивных производительности. Язык программирования Julia заполняет эту роль: это гибкий динамический язык, подходящий для научных и численных вычислений, с производительностью, сравнимой с традиционными статичными языками.

4 Выполнение лабораторной работы

Прежде всего, был запущен Jupyter Notebook с уже установленной на него Julia (?@fig-001).



```
Командная строка - jupyter
Microsoft Windows [Version 10.0.26100.6946]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\Home>jupyter notebook
[I 2025-09-13 14:00:40.102 LabApp] JupyterLab extension loaded from C:\Users\Home\AppData\Local\Programs\Python\Python310\Lib\site-packages\jupyterlab
[I 2025-09-13 14:00:40.103 LabApp] JupyterLab application directory is C:\Users\Home\AppData\Local\Programs\Python\Python310\share\jupyter\lab
[I 14:00:40.132 NotebookApp] Serving notebooks from local directory: C:\Users\Home
[I 14:00:40.132 NotebookApp] Jupyter Notebook 6.4.12 is running at:
[I 14:00:40.133 NotebookApp] http://localhost:8888/?token=b99a3b9a294e9396b7eb010a2dc1add86bc92647bfe93987
[I 14:00:40.135 NotebookApp] or http://127.0.0.1:8888/?token=b99a3b9a294e9396b7eb010a2dc1add86bc92647bfe93987
[I 14:00:40.136 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 14:00:40.501 NotebookApp]

To access the notebook, open this file in a browser:
file:///C:/Users/Home/AppData/Roaming/jupyter/runtime/nbserver-38068-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=b99a3b9a294e9396b7eb010a2dc1add86bc92647bfe93987
or http://127.0.0.1:8888/?token=b99a3b9a294e9396b7eb010a2dc1add86bc92647bfe93987
[I 14:01:45.856 NotebookApp] Creating new notebook in /Documents/РУДН/Компьютерный практикум по стохастическому анализу/Lab1
```

Далее, в рамках воспоминания основных команд – выведена дата и текущий пользователь (рис. 4.1).

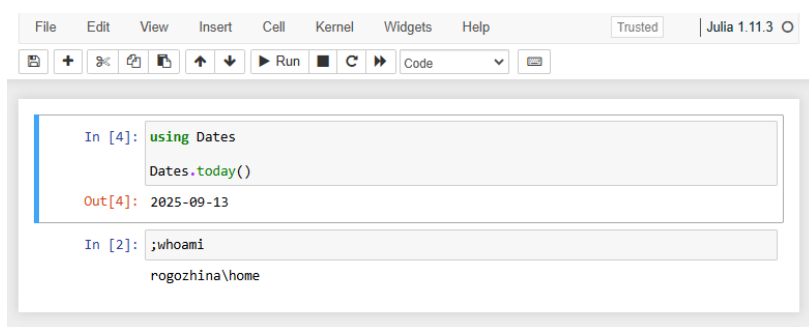


Рисунок 4.1: whoami

Изучите документацию по основным функциям Julia для чтения / записи / вывода информации на экран: `read()` (рис. 4.2), `readline()` и `readlines()`

(рис. 4.4), `readdlm()` (рис. 4.5), `print()` и `println()` (рис. 4.6), `show()` (рис. 4.7), `write()` ([100]. @fig-004]). Приведите свои примеры их использования, поясняя особенности их применения.

```

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 использования, поясняя особенности их применения.

In [18]: io = IOBuffer("JuliaLang is a GitHub organization");

In [15]: read(io, String)
Out[15]: "JuliaLang is a GitHub organization"

In [24]: io = IOBuffer("JuliaLang is a GitHub organization");

In [36]: path = "C:/Users/Home/Documents/РУДН/Компьютерный практикум по стохастическому анализу/lab1/myfile.txt"
Out[36]: "C:/Users/Home/Documents/РУДН/Компьютерный практикум по стохастическому анализу/lab1/myfile.txt"

In [19]: read(io, Char)
Out[19]: 'J': ASCII/Unicode U+004A (category Lu: Letter, uppercase)

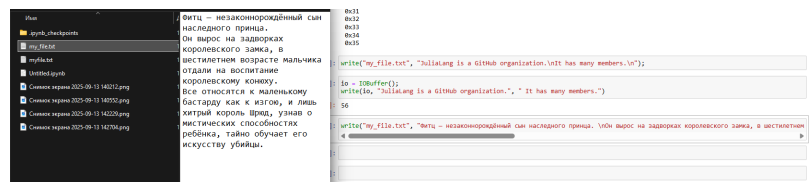
In [37]: read(path, String)
Out[37]: "12345678"

In [38]: read(path)
Out[38]: 8-element Vector{UInt8}:
 0x31
 0x32
 0x33
 0x34
 0x35
 0x36
 0x37
 0x38

In [39]: file = open(path, "r");
bytes = read(file, 5)
Out[39]: 5-element Vector{UInt8}:
 0x31
 0x32
 0x33
 0x34
 0x35

```

Рисунок 4.2: read + примеры



```

write("my_file.txt", "JuliaLang is a GitHub organization.txt has many members.\n");

io = IOBuffer();
write(io, "JuliaLang is a GitHub organization.", " It has many members.")
56

write("my_file.txt", "Фитц — незаконнорождённый сын наследного принца. Упоп вырос на задворках королевского замка, в шестилетнем
«

```

Рисунок 4.3: write + примеры

```

In [47]: readline("my_file.txt")
Out[47]: "Фитц — незаконнорождённый сын наследного принца."

In [51]: your_name = readline()
stdin> Miko green
Out[51]: "Miko green"

In [54]: readlines("my_file.txt")
Out[54]: 3-element Vector{String}:
 "Фитц — незаконнорождённый сын наследного принца."
 "Он вырос на задворках королевск" --- 101 bytes --- "спитание королевскому конюху."
 "Все относятся к маленькому баст" --- 167 bytes --- " обучает его искусству убийцы."

```

Рисунок 4.4: readline + readlines + примеры

```
In [56]: using DelimitedFiles

In [63]: readdlm("myfile.txt", ' ', Int, '\n')

Out[63]: 4x2 Matrix{Int64}:
 1  2
 3  4
 5  6
 7  8
```

```
In [64]: readdlm("myfile.txt", ' ', Float64, '\n')

Out[64]: 4x2 Matrix{Float64}:
 1.0  2.0
 3.0  4.0
 5.0  6.0
 7.0  8.0
```

Рисунок 4.5: readdlm + примеры

```
In [68]: print("hello")
          io = IOBuffer();
          print(io, "Hello", ' ', :World!)
          print(io, "have a good day!")
          String(take!(io))

hello

Out[68]: "Hello World!have a good day!"
```

```
In [69]: println("hello")
          io = IOBuffer();
          println(io, "Hello", ' ', :World!)
          println(io, "have a good day!")
          String(take!(io))

hello

Out[69]: "Hello World!\nhave a good day!\n"
```

Рисунок 4.6: print + println + примеры

```

In [75]: struct Day
          n::Int
        end

In [82]: Base.show(io::IO, ::MIME"text/plain", d::Day) = print(io, d.n, " days")

In [83]: Day(10)
Out[83]: 10 days

```

Рисунок 4.7: show + примеры

Изучите документацию по функции `parse()` (рис. 4.8). Приведите свои примеры её использования, поясняя особенности её применения.

2. Изучите документацию по функции `parse()`. Приведите свои примеры её использования, поясняя особенности её применения.

```

In [85]: parse{Int, "24082001", base=10}
Out[85]: 24082001

In [86]: parse{Int, "24082001", base=16}
Out[86]: 604512257

In [91]: parse{Int, "abc", base=10}
ArgumentError: invalid base 10 digit 'a' in "abc"

Stacktrace:
 [1] tryparse_internal(::Type{Int64}, s::String, startpos::Int64, endpos::Int64, base::Int64, raise::Bool)
   @ Base .\parse.jl:143
 [2] parse{::Type{Int64}, s::String; base::Int64}
   @ Base .\parse.jl:254
 [3] top-level scope
   @ In[91]:1

```

Рисунок 4.8: parse

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 (рис. 4.9).

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.

```
In [93]: a = 5
         b = 48
Out[93]: 48

In [94]: a+b
Out[94]: 53

In [95]: a-b
Out[95]: -43

In [96]: a*b
Out[96]: 240

In [97]: a/b
Out[97]: 0.10416666666666667

In [100]: b^a
Out[100]: 254803968

In [101]: sqrt(b)
Out[101]: 6.928203230275509

In [102]: a==b
Out[102]: false

In [103]: a<b
Out[103]: true
```

Рисунок 4.9: Основные математические операции

Приведите несколько своих примеров с пояснениями с операциями над матрицами и векторами: создание, сложение, вычитание, скалярное произведение, транспонирование, умножение на скаляр (рис. 4.10, рис. 4.11).

```
In [108]: a = 1
         b = 2
         c = 3
         d = 4
         e = 5
         f = 6
         h = 7
         i = 8
Out[108]: 8

In [109]: A = [a b; c d; e f; h i]
Out[109]: 4x2 Matrix{Int64}:
 1  2
 3  4
 5  6
 7  8

In [114]: A[3,2]
Out[114]: 6

In [115]: A^5
Out[115]: 4x2 Matrix{Int64}:
 5  10
15  20
25  30
35  40

In [116]: B = A*0.5
Out[116]: 4x2 Matrix{Float64}:
0.5  1.0
1.5  2.0
2.5  3.0
3.5  4.0
```

Рисунок 4.10: Создание матриц, умножение на скаляр

```
In [117]: A*5-B
Out[117]: 4x2 Matrix{Float64}:
  4.5  9.0
 13.5 18.0
 22.5 27.0
 31.5 36.0

In [120]: A*B'
Out[120]: 4x4 Matrix{Float64}:
  2.5  5.5  8.5 11.5
  5.5 12.5 19.5 26.5
  8.5 19.5 30.5 41.5
 11.5 26.5 41.5 56.5

In [121]: C = readlm("myfile.txt", ' ', Float64, '\n')
Out[121]: 4x2 Matrix{Float64}:
  1.0  2.0
  3.0  4.0
  5.0  6.0
  7.0  8.0
```

```
In [122]: C*B'
Out[122]: 4x4 Matrix{Float64}:
  2.5  5.5  8.5 11.5
  5.5 12.5 19.5 26.5
  8.5 19.5 30.5 41.5
 11.5 26.5 41.5 56.5
```

Рисунок 4.11: Операции с матрицами

5 Выводы

В ходе работы был установлен я.п. Julia на Jupyter Notebook, освоены основные функции любого я.п. (read, print, write, show), а также основные математические операции со скалярами, векторами и матрицами.

Список литературы