

# Лабораторная работа №1

Компьютерный практикум по статистическому анализу

Надежда Александровна Рогожина

2025-09-13

# Содержание I

1. Информация

2. Введение

3. Выполнение лабораторной работы

4. Выводы

# Раздел 1

## 1. Информация

## 1.1 Докладчик

► Рогожина Надежда Александровна

## 1.1 Докладчик

- ▶ Рогожина Надежда Александровна
- ▶ студентка 4 курса НФИбд-02-22

## 1.1 Докладчик

- ▶ Рогожина Надежда Александровна
- ▶ студентка 4 курса НФИбд-02-22
- ▶ Российский университет дружбы народов им. П. Лумумбы

## 1.1 Докладчик

- ▶ Рогожина Надежда Александровна
- ▶ студентка 4 курса НФИбд-02-22
- ▶ Российский университет дружбы народов им. П. Лумумбы
- ▶ 1132222840@rudn.ru

## 1.1 Докладчик

- ▶ Рогожина Надежда Александровна
- ▶ студентка 4 курса НФИбд-02-22
- ▶ Российский университет дружбы народов им. П. Лумумбы
- ▶ 1132222840@rudn.ru
- ▶ <https://MikoGreen.github.io/ru/>



## Раздел 2

### 2. Введение

## 2.1 Цель работы

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

## 2.2 Задание

1. Установите под свою операционную систему Julia, Jupyter (разделы 1.3.1 и 1.3.2).

## 2.2 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.

## 2.2 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).

## 2.3 Теоретическое введение

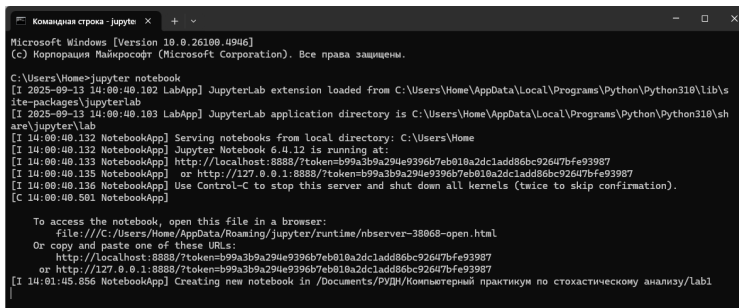
Научные вычисления традиционно требуют высочайшей производительности, однако эксперты по доменам в значительной степени перенесены на более медленные динамические языки для ежедневной работы. К счастью, методы современного языка и компилятора позволяют в основном устранить компромисс производительности 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.

## Раздел 3

### 3. Выполнение лабораторной работы

## 3.1 Подготовка

Прежде всего, был запущен Jupyter Notebook с уже установленной на него Julia.



```
Командная строка - jupyter x + v
Microsoft Windows [Version 10.0.26100.4946]
(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]

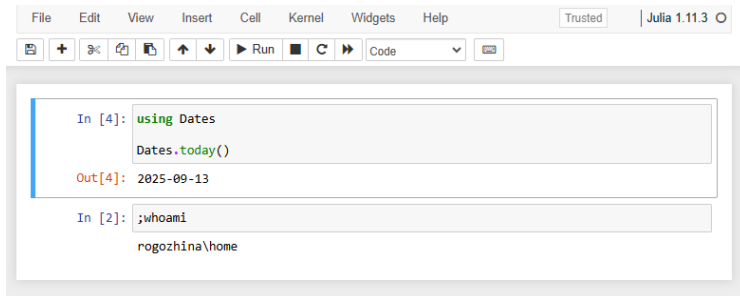
To access the notebook, open this file in a browser:
file:///C:/Users/Home/AppData/Roaming/jupyter/runtime/nbserver-38068-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=b99a3b9a294e9396b7eb010a2dc1add86bc92647bfe93987
or http://127.0.0.1:8888/?token=b99a3b9a294e9396b7eb010a2dc1add86bc92647bfe93987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

Рисунок 1: cmd



## 3.2 Подготовка

Далее, в рамках вспоминания основных команд – выведена дата и текущий пользователь.



```
File Edit View Insert Cell Kernel Widgets Help Trusted | Julia 1.11.3 ○  
[Icons] [Run] [Code]  
  
In [4]: using Dates  
        Dates.today()  
Out[4]: 2025-09-13  
  
In [2]: ;whoami  
        rogozhina\home
```

Рисунок 2: whoami

## 3.3 Задание 1

Изучите документацию по основным функциям Julia для чтения / записи / вывода информации на экран: `read()`, `readline()` и `readlines()`, `readdlm()`, `print()` и `println()`, `show()`, `write()`. Приведите свои примеры их использования, поясняя особенности их применения.

## 3.4 Задание 1

1. Изучите документацию по основным функциям Julia для чтения / записи / вывода информации на экран: `read()`, `readline()`, `readlines()`, `readall()`, `print()`, `println()`, `show()`, `write()`. Приведите свои примеры их 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/Homa/Documents/РЭД/У/Компьютерный практикум по стохастическому анализу/lab1/myfile.txt"
Out[36]: "C:/Users/Homa/Documents/РЭД/У/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 3: read + примеры

## 3.5 Задание 1

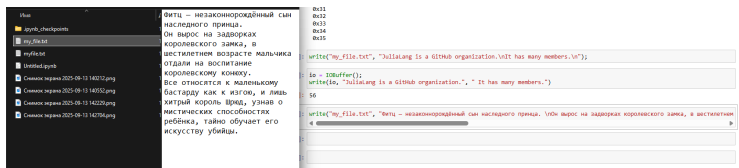


Рисунок 4: write + примеры

## 3.6 Задание 1

```
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" ... 181 bytes ... "спитание королевскому конюху. "
  "Все относятся к маленькому баст" ... 167 bytes ... " обучает его искусству убийцы."
```

Рисунок 5: readline + readlines + примеры

## 3.7 Задание 1

```
In [56]: using DelimitedFiles
```

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

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

```
In [64]: readlm("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 6: readlm + примеры

## 3.8 Задание 1

```
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 7: print + println + примеры

## 3.9 Задание 1

```
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 8: show + примеры



## 3.10 Задание 2

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

## 3.11 Задание 2

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 9: `parse`

## 3.12 Задание 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.

## 3.13 Задание 3

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 [98]: b*a  
Out[98]: 254883968  
  
In [99]: sqrt(b)  
Out[99]: 6.928203258275589  
  
In [100]: a==b  
Out[100]: false  
  
In [101]: a<b  
Out[101]: true
```

Рисунок 10: Основные математические операции

## 3.14 Задание 4

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

## 3.15 Задание 4

```
In [100]: a = 1  
          b = 2  
          c = 3  
          d = 4  
          e = 5  
          f = 6  
          h = 7  
          i = 8  
  
Out[100]: 8  
  
In [100]: A = [a b; c d; e f; h i]  
Out[100]: 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 [110]: B = A*0.5  
Out[110]: 4x2 Matrix{Float64}:  
          0.5 1.0  
          1.5 2.0  
          2.5 3.0  
          3.5 4.0
```

Рисунок 11: Создание матриц, умножение на скаляр

## 3.16 Задание 4

```
In [117]: A*S-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 12: Операции с матрицами

## Раздел 4

### 4. Выводы



## 4.1 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.