

Отчёт по лабораторной работе №1

дисциплина: Компьютерный практикум по статистическому анализу
данных

Студент: Кузнецова София Вадимовна

Содержание

Цель работы	5
Выполнение лабораторной работы	6
Подготовка инструментария к работе	6
Основы синтаксиса Julia на примерах	8
Самостоятельная работа	11
Вывод	15
Список литературы. Библиография	16

Список иллюстраций

0.1	Установка менеджера пакетов Chocolatey	6
0.2	Установка Far Manager	7
0.3	Установка Notepad++	7
0.4	Установка Julia	7
0.5	Установка Anaconda Distribution (Python 3.x)	8
0.6	Установка пакетов для работы с Jupyter	8
0.7	Примеры определения типа числовых величин	9
0.8	Примеры приведения аргументов к одному типу	9
0.9	Примеры определения функций	10
0.10	Примеры работы с массивами	10
0.11	Примеры работы с функциями для чтения/записи/вывода информа- ции на экран	11
0.12	Пример работы с функцией parse	11
0.13	Примеры работы базовых математических операций	12
0.14	Примеры работы базовых математических операций	13
0.15	Примеры работы с операциями над матрицами	14

Список таблиц

Цель работы

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

Выполнение лабораторной работы

Подготовка инструментария к работе

Так как мы используем ОС типа Windows для различных установок будем использовать менеджер пакетов Chocolatey (<https://chocolatey.org/>), который установим через Administrative Shell:

```
PS C:\Users\sofik> Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))
ПРЕДУПРЕЖДЕНИЕ: 'choco' was found at 'C:\ProgramData\chocolatey\bin\choco.exe'.
ПРЕДУПРЕЖДЕНИЕ: An existing Chocolatey installation was detected. Installation will not continue. This script will not overwrite existing installations.
If there is no Chocolatey installation at 'C:\ProgramData\chocolatey', delete the folder and attempt the installation again.

Please use choco upgrade chocolatey to handle upgrades of Chocolatey itself.
If the existing installation is not functional or a prior installation did not complete, follow these steps:
- Backup the files at the path listed above so you can restore your previous installation if needed.
- Remove the existing installation manually.
- Rerun this installation script.
- Reinstall any packages previously installed, if needed (refer to the lib folder in the backup).

Once installation is completed, the backup folder is no longer needed and can be deleted.
```

Рис. 0.1: Установка менеджера пакетов Chocolatey

Далее посредством установленного менеджера установим Far Manager, Notepad++, Julia, Anaconda Distribution (Python 3.x):

```
PS C:\Users\sofik> choco install far -y
Chocolatey v2.5.1
Installing the following packages:
far
By installing, you accept licenses for the packages.
far v3.0.6446 already installed.
Use --force to reinstall, specify a version to install, or try upgrade.

Chocolatey installed 0/1 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).

Warnings:
- far - far v3.0.6446 already installed.
Use --force to reinstall, specify a version to install, or try upgrade.
```

Рис. 0.2: Установка Far Manager

```
PS C:\Users\sofik> choco install notepadplusplus -y
Chocolatey v2.5.1
Installing the following packages:
notepadplusplus
By installing, you accept licenses for the packages.
notepadplusplus v8.8.5 already installed.
Use --force to reinstall, specify a version to install, or try upgrade.

Chocolatey installed 0/1 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).

Warnings:
- notepadplusplus - notepadplusplus v8.8.5 already installed.
Use --force to reinstall, specify a version to install, or try upgrade.
```

Рис. 0.3: Установка Notepad++

```
PS C:\Users\sofik> choco install julia -y
Chocolatey v2.5.1
Installing the following packages:
julia
By installing, you accept licenses for the packages.
julia v1.11.6 already installed.
Use --force to reinstall, specify a version to install, or try upgrade.

Chocolatey installed 0/1 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).

Warnings:
- julia - julia v1.11.6 already installed.
Use --force to reinstall, specify a version to install, or try upgrade.
```

Рис. 0.4: Установка Julia

```

PS C:\Users\sofik> choco install anaconda3 -y
Chocolatey v2.5.1
Installing the following packages:
anaconda3
By installing, you accept licenses for the packages.
anaconda3 v2025.6.0 already installed.
Use --force to reinstall, specify a version to install, or try upgrade.

Chocolatey installed 0/1 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).

Warnings:
- anaconda3 - anaconda3 v2025.6.0 already installed.
Use --force to reinstall, specify a version to install, or try upgrade.

```

Рис. 0.5: Установка Anaconda Distribution (Python 3.x)

Следующим шагом установим пакеты для работы с Jupyter. Для этого перейдём в пакетный режим Julia, нажав на клавиатуре знак закрывающейся квадратной скобки], затем введём add IJulia:

```

PS C:\Users\sofik> julia

┌───┐
│   │ Documentation: https://docs.julialang.org
│   │ Type "?" for help, "]?" for Pkg help.
│   │ Version 1.11.6 (2025-07-09)
│   │ Official https://julialang.org/ release
└───┘

(@v1.11) pkg> add IJulia
Resolving package versions...
No Changes to 'C:\Users\sofik\.julia\environments\v1.11\Project.toml'
No Changes to 'C:\Users\sofik\.julia\environments\v1.11\Manifest.toml'

(@v1.11) pkg>

```

Рис. 0.6: Установка пакетов для работы с Jupyter

Основы синтаксиса Julia на примерах

Для начала потренируемся с определением типов числовых величин:


```
[85]: typeof(3), typeof(3.5), typeof(3/3.5), typeof(sqrt(3+4im)), typeof(pi)
[85]: (Int64, Float64, Float64, ComplexF64, Irrational{::π})

[87]: 1.0/0.0, 1.0/(-0.0), 0.0/0.0
[87]: (Inf, -Inf, NaN)

[89]: typeof(1.0/0.0), typeof(1.0/-0.0), typeof(0.0/0.0)
[89]: (Float64, Float64, Float64)

[91]: 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]
```

Рис. 0.7: Примеры определения типа числовых величин

После чего приступим к рассмотрению приведения аргументов к одному типу:

```
[93]: Int64(2.0), Char(2), typeof(Char(2))
[93]: (2, '\x02', Char)

[95]: convert{Int64, 2.0}, convert{Char,2}
[95]: (2, '\x02')

[97]: typeof(promote{Int8(1), Float16(4.5), Float32(4.1)})
[97]: Tuple{Float32, Float32, Float32}
```

Рис. 0.8: Примеры приведения аргументов к одному типу

И рассмотрим примеры определения функций, а также работу с массивами:

```

[99]: function f(x)
      x^2
      end

[99]: f (generic function with 1 method)

[101]: f(4)

[101]: 16

[103]: g(x)=x^2

[103]: g (generic function with 1 method)

[105]: g(8)

[105]: 64

```

Рис. 0.9: Примеры определения функций

```

[107]: a = [4 7 6]
      b = [1, 2, 3]
      a[2], b[2]

[107]: (7, 2)

[109]: a = 1; b = 2; c = 3; d = 4
      Am = [a b; c d]

[109]: 2x2 Matrix{Int64}:
      1  2
      3  4

[111]: Am[1,1], Am[1,2], Am[2,1], Am[2,2]

[111]: (1, 2, 3, 4)

[115]: aa = [1 2]
      AA = [1 2; 3 4]
      aa*AA*aa'

[115]: 1x1 Matrix{Int64}:
      27

[117]: aa, AA, aa'

[117]: ([1 2], [1 2; 3 4], [1; 2; ;])

```

Рис. 0.10: Примеры работы с массивами

Самостоятельная работа

В первом задании мы рассмотрим основные функции для чтения / записи / вывода информации на экран. Для этого составим свои примеры:

```
[120]: write("myfile.txt", "Hello, world!\nMy name is Sofia.\n")
```

```
[120]: 32
```

```
[122]: io = open("myfile.txt", "r")  
read(io, String)
```

```
[122]: "Hello, world!\nMy name is Sofia.\n"
```

```
[125]: readline("myfile.txt")
```

```
[125]: "Hello, world!"
```

```
[128]: readlines("myfile.txt")
```

```
[128]: 2-element Vector{String}:  
      "Hello, world!"  
      "My name is Sofia."
```

```
[131]: print("Hello, World!")
```

```
Hello, World!
```

```
[133]: println("Hello", ",", " World!")
```

```
Hello, World!
```

```
[135]: show("Hello, World!")
```

```
"Hello, World!"
```

Рис. 0.11: Примеры работы с функциями для чтения/записи/вывода информации на экран

Во втором задании составим пример для функции `parse()`:

```
[141]: ex1 = Meta.parse("(4 + 4) / 2")
```

```
[141]: :((4 + 4) / 2)
```

Рис. 0.12: Пример работы с функцией `parse`

Далее изучим синтаксис Julia для базовых математических операций с разным типом переменных:

```
[143]: 1 + 1
[143]: 2

[145]: 5 - 1.0
[145]: 4.0

[147]: 2.0 + 3.0
[147]: 5.0

[149]: 2.0 * 3.0
[149]: 6.0

[151]: 6 / 2
[151]: 3.0

[153]: 7.0 % 6
[153]: 1.0

[155]: 2^5
[155]: 32
```

Рис. 0.13: Примеры работы базовых математических операций

```
[157]: 1 == 1
[157]: true

[159]: 2 != 2
[159]: false

[161]: 5.0 < 5
[161]: false

[165]: 3 > 2.7
[165]: true

[167]: x = true
      !x
[167]: false

[169]: x = true
      y = false
      x || y
[169]: true
```

Рис. 0.14: Примеры работы базовых математических операций

В конце работы приведём несколько примеров с операциями над матрицами:

```

[173]: using LinearAlgebra
       A = [1 2 3; 5 3 8; 7 9 3]

[173]: 3x3 Matrix{Int64}:
       1  2  3
       5  3  8
       7  9  3

[175]: det(A)

[175]: 91.0

[177]: tr(A)

[177]: 7

[179]: inv(A)

[179]: 3x3 Matrix{Float64}:
       -0.692308  0.230769  0.0769231
       0.450549 -0.197802  0.0769231
       0.263736  0.0549451 -0.0769231

[181]: B = [1 1 1; 1 1 1; 1 1 1]
       A - B

[181]: 3x3 Matrix{Int64}:
       0  1  2
       4  2  7
       6  8  2

[183]: A + B

[183]: 3x3 Matrix{Int64}:
       2  3  4
       6  4  9
       8 10  4

[185]: A * B

[185]: 3x3 Matrix{Int64}:
       6  6  6
      16 16 16
      19 19 19

```

Рис. 0.15: Примеры работы с операциями над матрицами

Вывод

В ходе выполнения лабораторной работы были получены навыки по подготовке рабочего пространства и инструментария для работы с языком программирования Julia, а также познакомились на простейших примерах с основами синтаксиса Julia.

Список литературы. Библиография

[1] Julia Documentation: <https://docs.julialang.org/en/v1/>