

Отчет по лабораторной работе №3

**Дисциплина: Компьютерный практикум по статистическому анализу
данных**

Лобанова Полина Иннокентьевна

Содержание

1	Цель работы	6
2	Задание	7
3	Выполнение лабораторной работы	8
4	Выводы	29
	Список литературы	30

Список иллюстраций

3.1	Примеры с циклами <i>while</i> и <i>for</i>	8
3.2	Примеры с циклами <i>while</i> и <i>for</i>	9
3.3	Примеры с циклами <i>while</i> и <i>for</i>	9
3.4	Примеры с условными выражениями	10
3.5	Примеры с функциями	10
3.6	Примеры с функциями	11
3.7	Примеры с функциями	11
3.8	Примеры с функциями	12
3.9	Примеры с функциями	12
3.10	Примеры с сторонними библиотеками	13
3.11	Примеры с сторонними библиотеками	13
3.12	Задание 1.1	14
3.13	Задание 1.1	15
3.14	Задание 1.2	16
3.15	Задание 1.3	17
3.16	Задание 2	17
3.17	Задание 3	18
3.18	Задание 4	18
3.19	Задача	18
3.20	Задание 5	19
3.21	Задание 6	20
3.22	Задача	20
3.23	Задание 7.1	21
3.24	Задание 7.2	21
3.25	Задание 7.3	22
3.26	Задание 7.4	22
3.27	Задание 8.1	23
3.28	Задача	23
3.29	Задание 8.2	24
3.30	Задание 8.3	24
3.31	Задание 8.4	25
3.32	Задача	25
3.33	Задание 9	25
3.34	Задание 10.1	26
3.35	Задание 10.2	26
3.36	Задание 10.3	27
3.37	Задача	27

3.38 Задание 11	28
---------------------------	----

Список таблиц

1 Цель работы

Основная цель работы — освоить применение циклов функций и сторонних для Julia пакетов для решения задач линейной алгебры и работы с матрицами.

2 Задание

1. Используя Jupyter Lab, повторите примеры из раздела 3.2.
2. Выполните задания для самостоятельной работы (раздел 3.4).

3 Выполнение лабораторной работы

1. Повторила примеры с циклами while и for.

[2]: # пока n<10 прибавить к n единицу и распечатать значение:

```
n=0
while n < 10
    n+=1
    println(n)
end
```

```
1
2
3
4
5
6
7
8
9
10
```

[4]: myfriends = ["Ted", "Robyn", "Barney", "Lily", "Marshall"]

```
i = 1
while i <= length(myfriends)
    friend= myfriends[i]
    println("Hi $friend, it's great to see you!")
    i+=1
end
```

```
Hi Ted, it's great to see you!
Hi Robyn, it's great to see you!
Hi Barney, it's great to see you!
Hi Lily, it's great to see you!
Hi Marshall, it's great to see you!
```

Рис. 3.1: Примеры с циклами while и for


```
[5]: for n in 1:2:10
      println(n)
    end

1
3
5
7
9

[6]: for friend in myfriends
      println("Hi $friend, it's great to see you!")
    end

Hi Ted, it's great to see you!
Hi Robyn, it's great to see you!
Hi Barney, it's great to see you!
Hi Lily, it's great to see you!
Hi Marshall, it's great to see you!

[8]: # инициализация массива m x n из нулей:
      m, n = 5, 5
      A = fill{0, (m,n)}
      # формирование массива, в котором значение каждой записи
      # является суммой индексов строки и столбца:
      for i in 1:m
        for j in 1:n
          A[i,j] = i+j
        end
      end
      A

[8]: 5x5 Matrix{Int64}:
      2  3  4  5  6
      3  4  5  6  7
      4  5  6  7  8
      5  6  7  8  9
      6  7  8  9 10
```

Рис. 3.2: Примеры с циклами *while* и *for*

```
[9]: B=fill{0, (m,n)}
      for i in 1:m, j in 1:n
        B[i, j] = i+j
      end
      B

[9]: 5x5 Matrix{Int64}:
      2  3  4  5  6
      3  4  5  6  7
      4  5  6  7  8
      5  6  7  8  9
      6  7  8  9 10

[10]: C = [i + j for i in 1:m, j in 1:n]
      C

[10]: 5x5 Matrix{Int64}:
      2  3  4  5  6
      3  4  5  6  7
      4  5  6  7  8
      5  6  7  8  9
      6  7  8  9 10
```

Рис. 3.3: Примеры с циклами *while* и *for*

2. Повторила примеры с условными выражениями.

```
[14]:
# используем `&&` для реализации операции "AND"
# операция % вычисляет остаток от деления
N=22
if (N % 3 == 0) && (N % 5 == 0)
  println("FizzBuzz")
elseif N % 3 == 0
  println("Fizz")
elseif N % 5 == 0
  println("Buzz")
else
  println(N)
end

22

[17]:
#Пример использования тернарного оператора:
x = 5
y = 10
(x>y) ? x : y

[17]:

10
```

Рис. 3.4: Примеры с условными выражениями

3. Повторила примеры с функциями.

```
[21]: function sayhi(name)
println("Hi $name, it's great to see you!")
end
# функция возведения в квадрат:
function f(x)
x^2
end
sayhi("polly")
f(12)

Hi polly, it's great to see you!

[21]: 144

[22]: sayhi2(name) = println("Hi $name, it's great to see you!")
f2(x) = x^2
sayhi2("C-3PO")
f2(42)

Hi C-3PO, it's great to see you!

[22]: 1764

[31]: sayhi3 = name -> println("Hi $name, it's great to see you!")
f3 = x -> x^2

[31]: #13 (generic function with 1 method)

[32]: f3(9)

[32]: 81
```

Рис. 3.5: Примеры с функциями

```
[36]: # задаём массив v:
v = [3, 5, 2]
sort(v)

[36]: 3-element Vector{Int64}:
      2
      3
      5

[37]: v

[37]: 3-element Vector{Int64}:
      3
      5
      2

[38]: sort!(v)

[38]: 3-element Vector{Int64}:
      2
      3
      5

[39]: v

[39]: 3-element Vector{Int64}:
      2
      3
      5
```

Рис. 3.6: Примеры с функциями

```
[40]: f(x) = x^2
      map(f, [1, 2, 3])

[40]: 3-element Vector{Int64}:
      1
      4
      9

[41]: map(x -> x^3, [1, 2, 3])

[41]: 3-element Vector{Int64}:
      1
      8
     27

[42]: broadcast(f, [1, 2, 3])

[42]: 3-element Vector{Int64}:
      1
      4
      9

[43]: f.([1, 2, 3])

[43]: 3-element Vector{Int64}:
      1
      4
      9

[44]: # задаём матрицу A:
      A = [i + 3*j for j in 0:2, i in 1:3]
```

Рис. 3.7: Примеры с функциями

```
[44]: # Задаём матрицу A:
A = [i + 3*j for j in 0:2, i in 1:3]

[44]: 3×3 Matrix{Int64}:
 1  2  3
 4  5  6
 7  8  9

[45]: # Вызываем функцию f возведения в квадрат
f(A)

[45]: 3×3 Matrix{Int64}:
 30  36  42
 66  81  96
102 126 150

[47]: B = f.(A)

[47]: 3×3 Matrix{Int64}:
 1  4  9
16 25 36
49 64 81
```

Рис. 3.8: Примеры с функциями

```
[48]: A .+2 .* f.(A) ./ A

[48]: 3×3 Matrix{Float64}:
 3.0  6.0  9.0
12.0 15.0 18.0
21.0 24.0 27.0

[49]: @. A + 2 * f(A) / A

[49]: 3×3 Matrix{Float64}:
 3.0  6.0  9.0
12.0 15.0 18.0
21.0 24.0 27.0

[50]: broadcast(x -> x + 2 * f(x) / x, A)

[50]: 3×3 Matrix{Float64}:
 3.0  6.0  9.0
12.0 15.0 18.0
21.0 24.0 27.0
```

Рис. 3.9: Примеры с функциями

4. Повторила примеры со сторонними библиотеками.

```
[51]: import Pkg
      Pkg.add("Example")


      Updating registry at `C:\Users\Nonna\.julia\registries\General.toml`
      Resolving package versions...
      Installed Example - v0.5.5
      Updating `C:\Users\Nonna\.julia\environments\v1.11\Project.toml`
      [7876af07] + Example v0.5.5
      Updating `C:\Users\Nonna\.julia\environments\v1.11\Manifest.toml`
      [7876af07] + Example v0.5.5
      Precompiling project...
      13219.1 ms ✓ Example
      1 dependency successfully precompiled in 43 seconds. 463 already precompiled.

[52]: Pkg.add("Colors")
      using Colors

      Resolving package versions...
      Installed Colors - v0.13.1
      Updating `C:\Users\Nonna\.julia\environments\v1.11\Project.toml`
      [5ae59095] + Colors v0.13.1
      Updating `C:\Users\Nonna\.julia\environments\v1.11\Manifest.toml`
      [5ae59095] ↑ Colors v0.13.0 ⇒ v0.13.1
      Precompiling project...
      4800.2 ms ✓ Colors
      1830.2 ms ✓ SparseMatrixColorings + SparseMatrixColoringsColorsExt
      3150.5 ms ✓ ColorSchemes
      8388.6 ms ✓ PlotUtils
      3083.3 ms ✓ PlotThemes
      3974.4 ms ✓ RecipesPipeline
      57268.9 ms ✓ Plots
      4806.4 ms ✓ Plots + UnitfulExt
      5099.7 ms ✓ Plots + JuliaExt
      9 dependencies successfully precompiled in 84 seconds. 455 already precompiled.
```

Рис. 3.10: Примеры с сторонними библиотеками

```
[64]: palette = distinguishable_colors(100)

[64]: 

[81]: rand(palette, 3, 3)


[81]: 
```

Рис. 3.11: Примеры с сторонними библиотеками

5. Используя циклы while и for:

– вывела на экран целые числа от 1 до 100 и напечатала их квадраты;

[86]:

```
#1.1
n = 1
while n <= 100
    println(n, " Квадрат: ", n^2)
    n+=1
end
```

```
1 Квадрат: 1
2 Квадрат: 4
3 Квадрат: 9
4 Квадрат: 16
5 Квадрат: 25
6 Квадрат: 36
7 Квадрат: 49
8 Квадрат: 64
9 Квадрат: 81
10 Квадрат: 100
11 Квадрат: 121
12 Квадрат: 144
13 Квадрат: 169
14 Квадрат: 196
15 Квадрат: 225
16 Квадрат: 256
17 Квадрат: 289
18 Квадрат: 324
19 Квадрат: 361
20 Квадрат: 400
21 Квадрат: 441
22 Квадрат: 484
23 Квадрат: 529
24 Квадрат: 576
25 Квадрат: 625
26 Квадрат: 676
27 Квадрат: 729
28 Квадрат: 784
29 Квадрат: 841
```

Рис. 3.12: Задание 1.1

```
[91]:  
n1 = 100  
for i in 1:n1  
    println(i, " Квадрат: ", i^2)  
end  
  
1 Квадрат: 1  
2 Квадрат: 4  
3 Квадрат: 9  
4 Квадрат: 16  
5 Квадрат: 25  
6 Квадрат: 36  
7 Квадрат: 49  
8 Квадрат: 64  
9 Квадрат: 81  
10 Квадрат: 100  
11 Квадрат: 121  
12 Квадрат: 144  
13 Квадрат: 169  
14 Квадрат: 196  
15 Квадрат: 225  
16 Квадрат: 256  
17 Квадрат: 289  
18 Квадрат: 324  
19 Квадрат: 361  
20 Квадрат: 400  
21 Квадрат: 441  
22 Квадрат: 484  
23 Квадрат: 529  
24 Квадрат: 576  
25 Квадрат: 625  
26 Квадрат: 676  
27 Квадрат: 729  
28 Квадрат: 784  
29 Квадрат: 841  
30 Квадрат: 900  
31 Квадрат: 961  
32 Квадрат: 1024  
33 Квадрат: 1089
```

Рис. 3.13: Задание 1.1

– создала словарь `squares`, который будет содержать целые числа в качестве ключей и квадраты в качестве их пар-значений;

```
#1.2
n = 1
squares = Dict()
while n<=10
    squares[n] = n^2
    n+=1
end
squares
```

[104]:

```
Dict{Any, Any} with 10 entries:
 5 => 25
 4 => 16
 6 => 36
 7 => 49
 2 => 4
10 => 100
 9 => 81
 8 => 64
 3 => 9
 1 => 1
```

[106]:

```
squares2=Dict()
for i in 1:10
    squares2[i]=i^2
end
squares2
```

[106]:

```
Dict{Any, Any} with 10 entries:
 5 => 25
 4 => 16
 6 => 36
 7 => 49
 2 => 4
```

Рис. 3.14: Задание 1.2

– создала массив `squares_arr`, содержащий квадраты всех чисел от 1 до 100.


```
•[107]:
#1.3
squares_arr = [i^2 for i in 1:100]
```

```
[107]:
100-element Vector{Int64}:
 1
 4
 9
16
25
36
49
64
81
100
121
144
169
 ⋮
7921
8100
8281
8464
8649
8836
9025
9216
9409
9604
9801
10000
```

Рис. 3.15: Задание 1.3

6. Написала условный оператор, который печатает число, если число чётное, и строку «нечётное», если число нечётное. Переписала код, используя тернарный оператор.

```
[112]:
#2
t = 101
if t % 2 == 0
    println(t)
else
    println("нечётное")
end

нечётное

[113]:
t%2==0 ? t : println("нечётное")

нечётное
```

Рис. 3.16: Задание 2

7. Написала функцию add_one, которая добавляет 1 к своему входу.

```
[114]:
#3
function add_one(x)
    x+=1
end
add_one(101)

[114]:
102
```

Рис. 3.17: Задание 3

8. Использовала `map()` или `broadcast()` для задания матрицы A , каждый элемент которой увеличивается на единицу по сравнению с предыдущим.

```
•[126]:
#4
A = fill(3, (3,3))
broadcast(x -> x+1, A)

[126]:
3x3 Matrix{Int64}:
 4  4  4
 4  4  4
 4  4  4
```

Рис. 3.18: Задание 4

9. Задала матрицу A следующего вида:

$$A = \begin{pmatrix} 1 & 1 & 3 \\ 5 & 2 & 6 \\ -2 & -1 & -3 \end{pmatrix}.$$

- Найдите A^3 .
- Замените третий столбец матрицы A на сумму второго и третьего столбцов.

Рис. 3.19: Задача

```
[135]:
#5
A = [[1 1 3]; [5 2 6]; [-2 -1 -3]]
function k(x)
    x^3
end
k(A)
```

```
[135]:
3×3 Matrix{Int64}:
 0  0  0
 0  0  0
 0  0  0
```

```
[136]:
for i in 1:3
    A[i, 3] = A[i, 1] + A[i, 2]
end
A
```

```
[136]:
3×3 Matrix{Int64}:
 1  1  2
 5  2  7
-2 -1 -3
```

Рис. 3.20: Задание 5

10. Создала матрицу B с элементами $B_{i1} = 10, B_{i2} = -10, B_{i3} = 10, i = 1, 2, \dots, 15$.
Вычислила матрицу $C = B^T B$.

```
[138]:
#6
B = fill(0, (15, 3))
for i in 1:15
    B[i, 1] = 10
    B[i, 2] = -10
    B[i, 3] = 10
end
B

[138]:
15x3 Matrix{Int64}:
 10  -10  10
 10  -10  10
 10  -10  10
 10  -10  10
 10  -10  10
 10  -10  10
 10  -10  10
 10  -10  10
 10  -10  10
 10  -10  10
 10  -10  10
 10  -10  10
 10  -10  10
 10  -10  10
 10  -10  10

[139]:
C = B' * B

[139]:
3x3 Matrix{Int64}:
1500  -1500  1500
-1500  1500  -1500
 1500  -1500  1500
```

Рис. 3.21: Задание 6

11. Создала матрицу Z размерности 6×6 , все элементы которой равны нулю, и матрицу E , все элементы которой равны 1. Используя цикл `while` или `for` и закономерности расположения элементов, создала следующие матрицы размерности 6×6 :

$$Z_1 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \quad Z_2 = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix},$$

$$Z_3 = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}, \quad Z_4 = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

Рис. 3.22: Задача

```
#7
Z = fill(0, (6, 6))
E = fill(1, (6, 6))
Z
```

[155]:

```
6×6 Matrix{Int64}:
 0  0  0  0  0  0
 0  0  0  0  0  0
 0  0  0  0  0  0
 0  0  0  0  0  0
 0  0  0  0  0  0
 0  0  0  0  0  0
```

[154]:

```
for i in 1:6, j in 1:6
    if j-i==1
        Z[i, j] = 1
    elseif i-j == 1
        Z[i, j] = 1
    else
        continue
    end
end
Z
```

[154]:

```
6×6 Matrix{Int64}:
 0  1  0  1  0  1
 1  0  1  0  1  0
 0  1  0  1  0  1
 1  0  1  0  1  0
 0  1  0  1  0  1
 1  0  1  0  1  0
```

Рис. 3.23: Задание 7.1

[151]:

```
for i in 1:6, j in 1:6
    if j-i == 2
        Z[i, j] = 1
    elseif i-j == 2
        Z[i, j] = 1
    elseif i==j
        Z[i, j] = 1
    else
        continue
    end
end
Z
```

[151]:

```
6×6 Matrix{Int64}:
 1  0  1  0  0  0
 0  1  0  1  0  0
 1  0  1  0  1  0
 0  1  0  1  0  1
 0  0  1  0  1  0
 0  0  0  1  0  1
```

Рис. 3.24: Задание 7.2

```
[153]:
for i in 1:6, j in 1:6
    if j+i == 5
        Z[i, j] = 1
    elseif i+j == 9
        Z[i, j] = 1
    elseif j+i == 7
        Z[i, j] = 1
    else
        continue
    end
end
Z
```

```
[153]:
6×6 Matrix{Int64}:
 0  0  0  1  0  1
 0  0  1  0  1  0
 0  1  0  1  0  1
 1  0  1  0  1  0
 0  1  0  1  0  0
 1  0  1  0  0  0
```

Рис. 3.25: Задание 7.3

```
•[157]:
for i in 1:6, j in 1:6
    if (i+j)%2==0
        Z[i, j] = 1
    else
        continue
    end
end
Z
```

```
[157]:
6×6 Matrix{Int64}:
 1  0  1  0  1  0
 0  1  0  1  0  1
 1  0  1  0  1  0
 0  1  0  1  0  1
 1  0  1  0  1  0
 0  1  0  1  0  1
```

Рис. 3.26: Задание 7.4

12. В языке R есть функция `outer()`. Фактически, это матричное умножение с возможностью изменить применяемую операцию (например, заменить произведение на сложение или возведение в степень).

– Написала свою функцию, аналогичную функции `outer()` языка R.

```

•[193]:
#8
function outer(x, y, operation)
    if operation == "*"
        c = x .* y'
    elseif operation == "+"
        c = x .+ y'
    elseif operation == "-"
        c = x .- y'
    elseif operation == "^"
        c = x .^ y'
    elseif operation == "/"
        if y != 0
            c = x ./ y'
        else
            println("невозможно")
        end
    elseif operation == "%"
        if y != 0
            c = x .% y'
        else
            println("невозможно")
        end
    end
end

x = [[1 2]; [3 4]]
y = [[5 6]; [7 8]]
outer(x, y, "%")

[193]:
2x2 Matrix{Int64}:
 1  2
 3  4

```

Рис. 3.27: Задание 8.1

– Используя написанную функцию `outer()`, создала матрицы следующей структуры:

$$A_1 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 4 & 5 & 6 \\ 3 & 4 & 5 & 6 & 7 \\ 4 & 5 & 6 & 7 & 8 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 2 & 4 & 8 & 16 & 32 \\ 3 & 9 & 27 & 81 & 243 \\ 4 & 16 & 64 & 256 & 1024 \end{pmatrix}.$$

$$A_3 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 0 \\ 2 & 3 & 4 & 0 & 1 \\ 3 & 4 & 0 & 1 & 2 \\ 4 & 0 & 1 & 2 & 3 \end{pmatrix}, \quad A_4 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 8 & 9 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 9 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{pmatrix},$$

Рис. 3.28: Задача

```
[174]:
A = [[-1 0 1 2 3];
      [0 1 2 3 4];
      [1 2 3 4 5];
      [2 3 4 5 6];
      [3 4 5 6 7]]

[174]:
5x5 Matrix{Int64}:
-1  0  1  2  3
 0  1  2  3  4
 1  2  3  4  5
 2  3  4  5  6
 3  4  5  6  7

[176]:
A1 = outer(A, 1, "+")

[176]:
5x5 Matrix{Int64}:
0  1  2  3  4
1  2  3  4  5
2  3  4  5  6
3  4  5  6  7
4  5  6  7  8
```

Рис. 3.29: Задание 8.2

```
[178]:
A2 = [outer((i-1), j, "^") for i in 1:5, j in 1:5]

[178]:
5x5 Matrix{Int64}:
0  0  0  0  0
1  1  1  1  1
2  4  8  16  32
3  9  27  81  243
4  16  64  256  1024

[196]:
a1 = outer(0:4, 0:4, "+")
A3 = outer(a1, 5, "%")

[196]:
5x5 Matrix{Int64}:
0  1  2  3  4
1  2  3  4  0
2  3  4  0  1
3  4  0  1  2
4  0  1  2  3
```

Рис. 3.30: Задание 8.3


```
[213]:
a2 = outer(0:9, 0:9, "+")
A4 = outer(a2, 10, "%")
A4
```

```
[213]:
10x10 Matrix{Int64}:
 0  1  2  3  4  5  6  7  8  9
 1  2  3  4  5  6  7  8  9  0
 2  3  4  5  6  7  8  9  0  1
 3  4  5  6  7  8  9  0  1  2
 4  5  6  7  8  9  0  1  2  3
 5  6  7  8  9  0  1  2  3  4
 6  7  8  9  0  1  2  3  4  5
 7  8  9  0  1  2  3  4  5  6
 8  9  0  1  2  3  4  5  6  7
 9  0  1  2  3  4  5  6  7  8
```

Рис. 3.31: Задание 8.4

13. Решила следующую систему линейных уравнений с 5 неизвестными:

$$\begin{cases} x_1 + 2x_2 + 3x_3 + 4x_4 + 5x_5 = 7, \\ 2x_1 + x_2 + 2x_3 + 3x_4 + 4x_5 = -1, \\ 3x_1 + 2x_2 + x_3 + 2x_4 + 3x_5 = -3, \\ 4x_1 + 3x_2 + 2x_3 + x_4 + 2x_5 = 5, \\ 5x_1 + 4x_2 + 3x_3 + 2x_4 + x_5 = 17, \end{cases}$$

Рис. 3.32: Задача

```
[256]:
#9
A = [[1 2 3 4 5];
      [2 1 2 3 4];
      [3 2 1 2 3];
      [4 3 2 1 2];
      [5 4 3 2 1]]

y = [7, -1, -3, 5, 17]

x = A^(-1) * y
```

```
[256]:
5-element Vector{Float64}:
-1.9999999999999996
 3.0
 5.0
 2.0
-4.0
```

Рис. 3.33: Задание 9

14. Создала матрицу М размерности 6×10 , элементами которой являются целые числа, выбранные случайным образом с повторениями из совокупности $1, 2, \dots, 10$.

– Нашла число элементов в каждой строке матрицы M, которые больше числа N (например, N = 4).

```
[257]:
#10
M = rand(1:10, (6, 10))

[257]:
6×10 Matrix{Int64}:
 9  1  5  5 10  9 10  6  5 10
 2  1  9  3  9  5  8  1  4  6
 3  4  3  1  7  5  8  8  5  2
 2  5  3  9  1  6  9  6  9  1
10  6  7  5  5  9  3  3  9  4
 7 10 10  1  4  4  8  4  2  3

[289]:
N=4
for i in 1:6
    count = 0
    for j in 1:10
        if M[i, j] > N
            count+=1
        else
            continue
        end
    end
    println(i, " ", count)
end

1 9
2 5
3 5
4 6
5 7
6 4
```

Рис. 3.34: Задание 10.1

– Определила, в каких строках матрицы M число M1 (например, M1 = 10) встречается ровно 2 раза.

```
[263]:
M1 = 10
for i in 1:6
    count = 0
    for j in 1:10
        if M[i, j] == M1
            count+=1
        else
            continue
        end
    end
    if count==2
        println(i, '\n')
    end
end

6
```

Рис. 3.35: Задание 10.2

– Определила все пары столбцов матрицы M, сумма элементов которых больше

K (например, K = 75).

```
[283]:  
K=75  
function coun(j)  
    c = 0  
    for i in 1:6  
        c+=M[i, j]  
    end  
    return c  
end  
  
for i in 1:10, t in 1:10  
    if i!=t && coun(i)+coun(t) > 75  
        println("Столбцы: ", i, ' ', t)  
    end  
end  
  
Столбцы: 1 7  
Столбцы: 3 7  
Столбцы: 5 7  
Столбцы: 6 7  
Столбцы: 7 1  
Столбцы: 7 3  
Столбцы: 7 5  
Столбцы: 7 6  
Столбцы: 7 9  
Столбцы: 9 7
```

Рис. 3.36: Задание 10.3

15. Вычислила:

$$\begin{aligned} & - \sum_{i=1}^{20} \sum_{j=1}^5 \frac{i^4}{(3+j)}, \\ & - \sum_{i=1}^{20} \sum_{j=1}^5 \frac{i^4}{(3+ij)}. \end{aligned}$$

Рис. 3.37: Задача

[290]:

```
#11
s1 = 0
for i in 1:20
    for j in 1:5
        s1 += i^4 / (3+j)
    end
    s1 = -s1
end
s1
```

[290]:

-77829.25

[288]:

```
s2 = 0
for i in 1:20
    for j in 1:5
        s2 += i^4 / (3+ i*j)
    end
    s2 = -s2
end
s2
```

[288]:

-8992.661163004545

Рис. 3.38: Задание 11

4 Выводы

Я освоила применение циклов функций и сторонних для Julia пакетов для решения задач линейной алгебры и работы с матрицами.

Список литературы