Лабораторная работа №2

Компьютерный практикум по статистическому анализу данных

Амуничников Антон Игоревич

2025-09-27

Содержание i

1. Информация

1.1 Докладчик

- Амуничников Антон Игоревич
- Группа: НПИбд-01-22
- Российский университет дружбы народов им. П. Лумумбы
- 1132227133@pfur.ru

2. Вводная часть

2.1 Цель работы

- Основная цель работы – изучить несколько структур данных, реализованных в Julia, научиться применять их и операции над ними для решения задач.

2.2 Задание

1. Используя Jupyter Lab, повторите примеры.
2. Выполните задания для самостоятельной работы.

3. Выполнение лабораторной работы

3.1 Кортеж

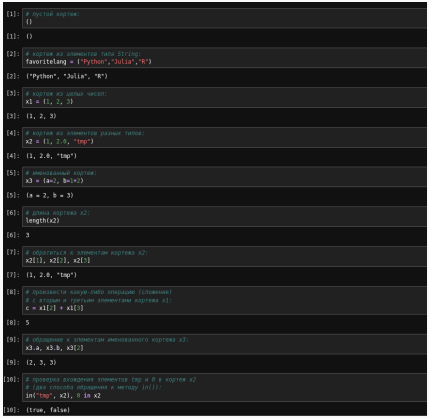

Рисунок 1: Примеры использования кортежей

3.2 Словарь

```
UndefVarError: `Словари` not defined in `Main`
Suggestion: check for spelling errors or missing imports.

[12]: # создать словарь с именем phonebook:
      phonebook = Dict("Иванов И.И." => ("867-5309","333-5544"), "Бухгалтерия" => "555-2368")

[12]: Dict{String, Any} with 2 entries:
        "Бухгалтерия" => "555-2368"
        "Иванов И.И." => ("867-5309", "333-5544")

[13]: # вывести ключи словаря:
      keys(phonebook)

[13]: KeySet for a Dict{String, Any} with 2 entries. Keys:
        "Бухгалтерия"
        "Иванов И.И."

[14]: # вывести значения элементов словаря:
      values(phonebook)

[14]: ValueIterator for a Dict{String, Any} with 2 entries. Values:
        "555-2368"
        ("867-5309", "333-5544")

[15]: # вывести заданные в словаре пары 'ключ - значение':
      pairs(phonebook)

[15]: Dict{String, Any} with 2 entries:
        "Бухгалтерия" => "555-2368"
        "Иванов И.И." => ("867-5309", "333-5544")

[16]: # проверка вхождения ключа в словарь:
      haskey(phonebook, "Иванов И.И.")

[16]: true

[17]: # добавить элемент в словарь:
      phonebook["Сидоров П.С."] = "555-3344"

[17]: "555-3344"

[19]: # удалить ключ и связанные с ним значения из словаря
      pop!(phonebook, "Иванов И.И.")

[19]: ("867-5309", "333-5544")

[20]: # Объединение словарей (функция merge()):
      a = Dict("foo" => 0.0, "bar" => 42.0);
      b = Dict("baz" => 17, "bar" => 13.0);
      merge(a, b), merge(b,a)

[20]: (Dict{String, Real}("bar" => 13.0, "baz" => 17, "foo" => 0.0), Dict{String, Real}("bar" => 42.0, "baz" => 17, "foo" => 0.0))
```

Рисунок 2: Примеры использования словарей

3.3 Множество

```
[21]: # создать множество из четырёх целочисленных значений:
      A = Set([1, 3, 4, 5])

[21]: Set{Int64} with 4 elements:
        5
        4
        3
        1

[22]: # создать множество из 11 символьных значений:
      B = Set("abrakadabra")

[22]: Set{Char} with 5 elements:
        'a'
        'd'
        'r'
        'k'
        'b'

[23]: # проверка эквивалентности двух множеств:
      S1 = Set([1,2]);
      S2 = Set([3,4]);
      issetequal(S1,S2)
      S3 = Set([1,2,2,1,1,2,1,2,1]);
      S4 = Set([2,1,1]);
      issetequal(S3,S4)

[23]: true

[24]: # проверка эквивалентности двух множеств:
      S1 = Set([1,2]);
      S2 = Set([3,4]);
      issetequal(S1,S2)

[24]: false

[25]: C=union(S1,S2)

[25]: Set{Int64} with 4 elements:
        4
        2
        3
        1

[26]: # пересечение множеств:
      D = intersect(S1,S3)

[26]: Set{Int64} with 2 elements:
        2
        1

[27]: # разность множеств:
      E = setdiff(S3,S1)
```

Рисунок 3: Примеры использования множеств

3.4 Массив

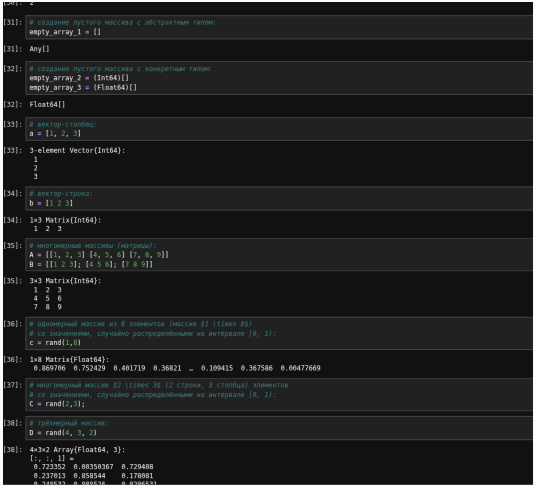

Рисунок 4: Примеры использования массивов

3.5 Массив

```
C = rand(2,6)
```

```
[36]: 1×8 Matrix{Float64}:
       0.869706  0.752429  0.401719  0.36821  …  0.109415  0.367586  0.00477669
```

```
[37]: # многомерный массив $2 \times 3$ (2 строки, 3 столбца) элементов
      # со значениями, случайно распределёнными на интервале [0, 1):
      C = rand(2,3);
```

```
[38]: # трёхмерный массив:
      D = rand(4, 3, 2)
```

```
[38]: 4×3×2 Array{Float64, 3}:
      [:, :, 1] =
       0.723352  0.00350367  0.729408
       0.237013  0.858544    0.178081
       0.248532  0.808526    0.0206531
       0.436335  0.981371    0.58473

      [:, :, 2] =
       0.660884  0.815358  0.439154
       0.525835  0.825208  0.863703
       0.847101  0.310867  0.328708
       0.642303  0.974095  0.305427
```

```
[39]: # массив из квадратных корней всех целых чисел от 1 до 10:
      roots = [sqrt(i) for i in 1:10]
```

```
[39]: 10-element Vector{Float64}:
       1.0
       1.4142135623730951
       1.7320508075688772
       2.0
       2.23606797749979
       2.449489742783178
       2.6457513110645907
       2.8284271247461903
       3.0
       3.1622776601683795
```

Рисунок 5: Примеры использования массивов

3.6 Массив

```
[40]: ar_1 = [3*i^2 for i in 1:2:9]
```

```
[40]: 5-element Vector{Int64}:
        3
       27
       75
      147
      243
```

```
[41]: # массив квадратов элементов, если квадрат не делится на 5 или 4:
      ar_2=[i^2 for i=1:10 if (i^2%5!=0 && i^2%4!=0)]
```

```
[41]: 4-element Vector{Int64}:
        1
        9
       49
       81
```

```
[42]: # одномерный массив из пяти единиц:
      ones(5)
```

```
[42]: 5-element Vector{Float64}:
       1.0
       1.0
       1.0
       1.0
       1.0
```

```
[43]: # двумерный массив 2x3 из единиц:
      ones(2,3)
```

```
[43]: 2×3 Matrix{Float64}:
       1.0  1.0  1.0
       1.0  1.0  1.0
```

```
[44]: # одномерный массив из 4 нулей:
      zeros(4)
```

```
[44]: 4-element Vector{Float64}:
       0.0
       0.0
       0.0
       0.0
```

```
[45]: # заполнить массив 3x2 цифрами 3.5
      fill(3.5,(3,2))
```

```
[45]: 3×2 Matrix{Float64}:
       3.5  3.5
       3.5  3.5
       3.5  3.5
```

3.7 Массив

```
[48]: # транспонирование
      b'
```

```
[48]: 6×2 adjoint(::Matrix{Int64}) with eltype Int64:
       1   2
       3   4
       5   6
       7   8
       9  10
      11  12
```

```
[49]: # транспонирование
      c = transpose(b)
```

```
[49]: 6×2 transpose(::Matrix{Int64}) with eltype Int64:
       1   2
       3   4
       5   6
       7   8
       9  10
      11  12
```

```
[50]: # массив 10x5 целых чисел в диапазоне [10, 20]:
      ar = rand(10:20, 10, 5)
```

```
[50]: 10×5 Matrix{Int64}:
      12  14  14  12  12
      16  12  16  13  19
      17  15  20  12  16
      19  16  17  15  20
      12  13  20  20  10
      20  19  15  18  19
      20  13  10  11  14
      19  19  13  14  15
      15  13  17  17  20
      14  18  14  11  20
```

```
[51]: # выбор всех значений строки в столбце 2:
      ar[:, 2]
```

```
[51]: 10-element Vector{Int64}:
      14
      12
      15
      16
      13
      19
      13
      19
      13
      18
```

3.8 Массив

```
      15  18  19  19  20
      10  11  13  14  20
      13  14  15  19  19
      13  15  17  17  20
      11  14  14  18  20
```

```
[58]: # поэлементное сравнение с числом
      # (результат - массив логических значений):
      ar .> 14
```

```
[58]: 10×5 BitMatrix:
      0  0  0  0  0
      1  0  1  0  1
      1  1  1  0  1
      1  1  1  1  1
      0  0  1  1  0
      1  1  1  1  1
      1  0  0  0  0
      1  1  0  0  1
      1  0  1  1  1
      0  1  0  0  1
```

```
[59]: # возврат индексов элементов массива, удовлетворящих условию:
      findall(ar .> 14)
```

```
[59]: 29-element Vector{CartesianIndex{2}}:
      CartesianIndex(2, 1)
      CartesianIndex(3, 1)
      CartesianIndex(4, 1)
      CartesianIndex(6, 1)
      CartesianIndex(7, 1)
      CartesianIndex(8, 1)
      CartesianIndex(9, 1)
      CartesianIndex(3, 2)
      CartesianIndex(4, 2)
      CartesianIndex(6, 2)
      CartesianIndex(8, 2)
      CartesianIndex(10, 2)
      CartesianIndex(2, 3)
      ⋮
      CartesianIndex(9, 3)
      CartesianIndex(4, 4)
      CartesianIndex(5, 4)
      CartesianIndex(6, 4)
      CartesianIndex(9, 4)
      CartesianIndex(2, 5)
      CartesianIndex(3, 5)
      CartesianIndex(4, 5)
      CartesianIndex(6, 5)
      CartesianIndex(8, 5)
      CartesianIndex(9, 5)
      CartesianIndex(10, 5)
```

3.9 Задание №1

Даны множества: $A = 0, 3, 4, 9, B = 1, 3, 4, 7, C = 0, 1, 2, 4, 7, 8, 9$. Найдем $P = A \cap B \cup A \cap B \cup A \cap C \cup B \cap C$.

```
[60]: #Задание 1
      A = Set([0,3,4,9])
      B = Set([1,3,4,7])
      C = Set([0,1,2,4,7,8,9])

      union(intersect(A,B), intersect(A,C), intersect(B,C))

[60]: Set{Int64} with 6 elements:
        0
        4
        7
        9
        3
        1
```

Рисунок 9: Задание №1. Работа с множествами

3.10 Задание №2

```
[61]: #Задание 2
      A = Set(["Иванов", "Петров", "Сидоров"])
      B = Set(["Иванов", "Птушкин"])#обьединение
      print(union(A,B))

      Set(["Иванов", "Петров", "Птушкин", "Сидоров"])

[62]: Set1 = Set([1, 2, 3, "Hello"])

[62]: Set{Any} with 4 elements:
        2
        "Hello"
        3
        1

[63]: println("\nElements of set:")
      for i in Set1
          println(i)
      end

      Elements of set:
      2
      Hello
      3
      1

[64]: print(in("Hello", Set1))

      true

[66]: Set1 = push!(Set1, "World")
      println("\nSet after adding one element: \n", Set1)

      Set after adding one element:
      Set(Any[2, "Hello", "World", 3, 1])

[68]: for i in 1:5
          push!(Set1, i)
      end
          println("\nSet after adding range of elements:\n", Set1)

      Set after adding range of elements:
```

3.11 Задание №3

```
[72]: #Задание 3
      b_1= [i for i in 1:21]
      print(b_1, '\n')

      [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21]

[75]: c = []
      c_1 = reverse([i for i in 1:21])
      print(c_1, '\n')

      [21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1]

[78]: print(vcat(b_1, c_1), '\n')

      [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1]

[80]: tmp = [4 6 3]
      println(tmp)

      [4 6 3]

[81]: println(vcat(fill.(tmp, [10 2 1])...))

      [4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 6, 6, 3]

[82]: println(vcat(fill.(tmp, [10 10 10])...))

      [4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3]

[83]: println(vcat(fill.(tmp, [11 10 10])...))

      [4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3]

[85]: println(vcat(fill.(tmp, [10 20 30])...))

      [4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
      3, 3, 3]
```

Рисунок 11: Задание №3. Работа с массивами

3.12 Задание №3

```
[86]: S8 = vcat(fill(2^tmp[1], 1), fill(2^tmp[2], 1), fill(2^tmp[3], 4))
      count_6 = count(x -> x == 6, S8)
      print(S8, '\n', count_6)

      [16, 64, 8, 8, 8, 8]
      0

[87]: # 3.10
      using Statistics
      f(x) = exp(x)*cos(x) #заменяю y на f так как такая функция уже создавалась
      Y = [f(x) for x in 3:0.1:6]
      mean(Y)

[87]: 53.11374594642971

[88]: # 3.11
      x_vals = [0.1^i for i in 3:3:36]
      y_vals = [0.2^j for j in 1:3:34]
      arr11 = vcat(x_vals, y_vals)

[88]: 24-element Vector{Float64}:
       0.001000000000000002
       1.0000000000000004e-6
       1.0000000000000005e-9
       1.0000000000000006e-12
       1.0000000000000009e-15
       1.000000000000001e-18
       1.0000000000000012e-21
       1.0000000000000014e-24
       1.0000000000000015e-27
       1.0000000000000017e-30
       1.0000000000000018e-33
       1.000000000000002e-36
       0.2
       0.0016000000000000003
       1.2800000000000005e-5
       1.0240000000000006e-7
       8.192000000000005e-10
       6.5536000000000055e-12
       5.242880000000056e-14
       4.194304000000056e-16
       3.3554432000000048e-18
       2.684354560000004e-20
       2.1474836480000035e-22
       1.717986918400003e-24
```

3.13 Задание №3

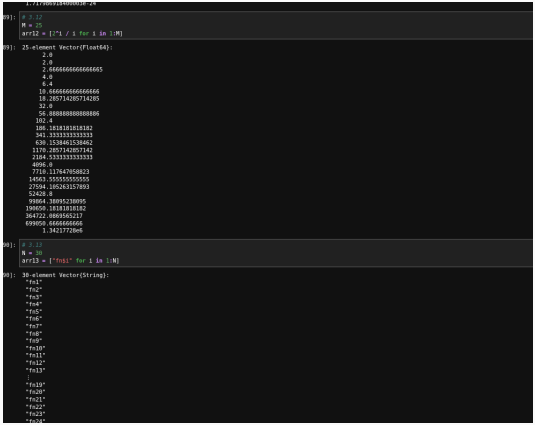

Рисунок 13: Задание №3. Работа с массивами

3.14 Задание №3

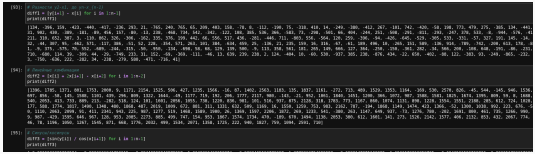

Рисунок 14: Задание №3. Работа с массивами

3.15 Задание №3

Рисунок 15: Задание №3. Работа с массивами

3.16 Задание №3

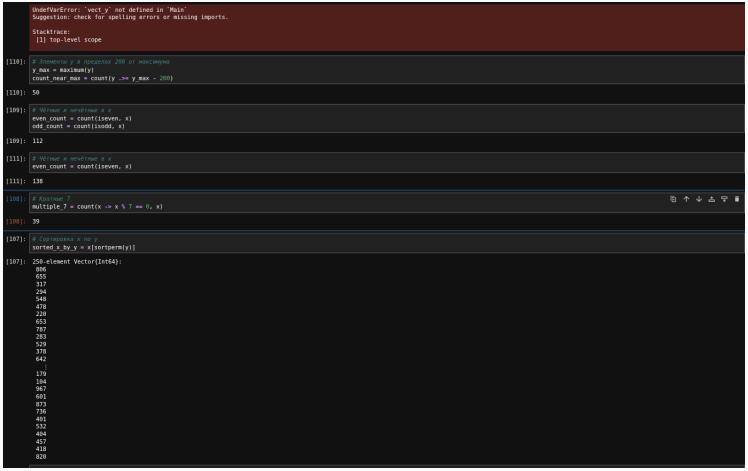

Рисунок 16: Задание №3. Работа с массивами

3.17 Задание №3

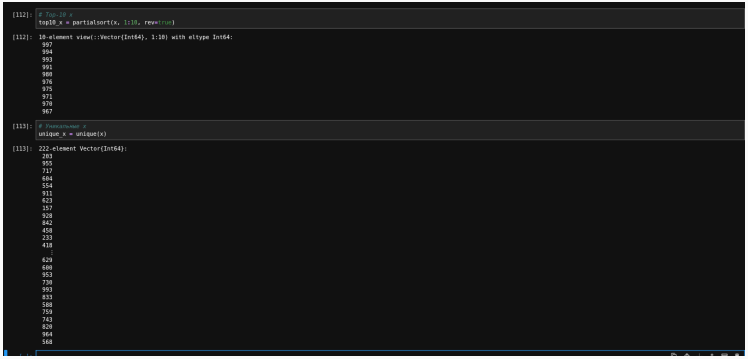

Рисунок 17: Задание №3. Работа с массивами

3.18 Задание №4

Создадим массив squares, в котором будут храниться квадраты всех целых чисел от 1 до 100.

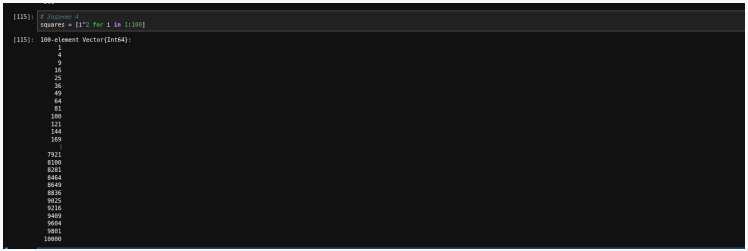

Рисунок 18: Задание №4

3.19 Задание №5

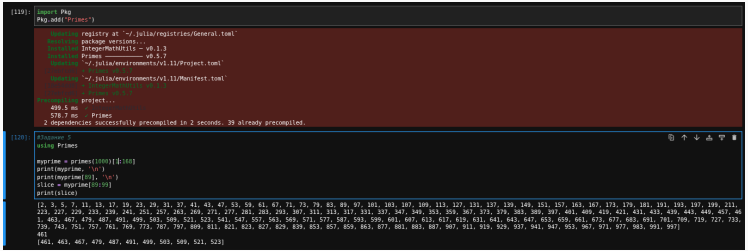

Рисунок 19: Задание №5. Работа с пакетом Primes

3.20 Задание №6

Вычислим следующие выражения.

$$\sum_{i=10}^{100}(i^3 + 4i^2);$$

$$\sum_{i=1}^{M}(\frac{2^i}{i} + \frac{3^i}{i^2}), M = 25;$$

$$1 + \frac{2}{3} + \left(\frac{2}{3}\frac{4}{5}\right) + \left(\frac{2}{3}\frac{4}{5}\frac{6}{7}\right) + \cdots + \left(\frac{2}{3}\frac{4}{5}\cdots\frac{39}{39}\right).$$

3.21 Задание №6

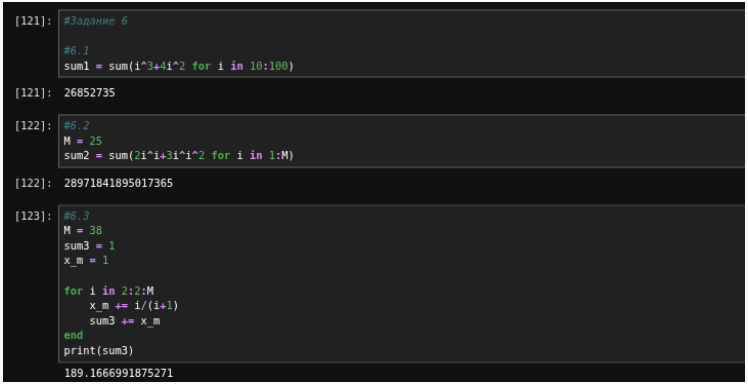

Рисунок 20: Задание №6

3.22 Выводы

В результате выполнения данной лабораторной работы я изучил несколько структур дыннх, реализованных в Julia, Научился применять их и операции