

# Лабораторная работа №2

Компьютерный практикум по статистическому анализу данных

---

Канева Екатерина, НФИбд-02-22

27 сентября 2025

Российский университет дружбы народов, Москва, Россия

## Информация

---

- Канева Екатерина Павловна
- студент группы НФИбд-02-22
- Российский университет дружбы народов
- 1132222004@rudn.ru
- <https://nevseros.github.io/ru/>

## Вводная часть

---

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

- Используя Jupyter Lab, повторить примеры.
- Выполнить задания для самостоятельной работы.

## Выполнение работы

---

Выполнила примеры с кортежами:

```
[31]: # именованный кортеж:
x1 = {a=2, b=1+2}

[32]: (a = 2, b = 3)

[33]: # длина кортежа x2:
length(x2)

[34]: 3

[35]: x2[1], x2[2], x2[3]

[36]: (1, 2.0, "tmp")

[37]: # произвести какую-либо операцию (сложение)
# c - вторая и третья элементы кортежа x1:
c = x1[2] + x1[3]

[38]: 5

[39]: # обращение к элементам именованного кортежа x1:
x1.a, x1.b, x1[2]

[40]: (2, 3, 3)
```

Рис. 1: Примеры с кортежами.



Выполнила примеры со словарями:

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

[27]: ValueIterator for a Dict[String, Any] with 2 entries. Values:
      "555-2368"
      ("667-5509", "333-5544")

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

[28]: Dict[String, Any] with 2 entries:
      "Бухгалтерия" => "555-2368"
      "Иванов И.И." => ("667-5509", "333-5544")

[31]: # проверить наличие ключа в словаре:
      haskey(phonebook, "Иванов И.И.")

[31]: true

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

[47]: Dict[String, Any] with 3 entries:
      "Сидоров П.С." => "555-3344"
      "Бухгалтерия" => "555-2368"
      "Иванов И.И." => ("667-5509", "333-5544")
```

Рис. 2: Примеры со словарями.

Выполнила примеры со множествами:

```
[329]: # объединение множеств:  
C = union(S1,S2)  
print(C)  
  
Set([4, 2, 3, 1])  
  
[331]: # пересечение множеств:  
D = intersect(S1,S3)  
print(D)  
  
Set([2, 1])  
  
[333]: # разность множеств:  
E = setdiff(S3,S1)  
  
[335]: Set{Int64} with 1 element:  
3  
  
[339]: # проверка владения элементов одного множества в другом:  
issubset(S1,S4)  
  
[340]: true  
  
[337]: # добавление элемента в множество:  
push!(S4, 99)  
print(S4)  
  
Set([2, 99, 3, 1])
```

Рис. 3: Примеры со множествами.

Выполнила примеры с массивами:

```
[100]: # примерный массив:
# = rand(4, 3, 2)

[101]: 4x3d Array{Float64, 3}:
[:, 1, 1] =
 0.235321  0.43721  0.4675089
 0.365003  0.462678  0.235354
 0.43863  0.443639  0.888912
 0.792592  0.38676  0.823595

[:, 1, 2] =
 0.385917  0.833856  0.547395
 0.289329  0.139213  0.97587
 0.807778  0.136123  0.108487
 0.339149  0.688889  0.395739

[102]: # массив из случайных корней трех целых чисел от 1 до 10:
roots = [sqrt(i) for i in 1:10]
println(roots)

[103]: [1.0, 1.4142135623730951, 1.7320508075688772, 2.0, 2.23606797749979, 2.449489742781178, 2.6457513110645967, 2.8284271247461905, 3.0, 3.1622776601683795]

[104]: # массив с элементами типа Float32,
# где x - целочисленные числа от 1 до 9 (вычисляются)
m_1 = [x^1.2 for i in 1:9]
println(m_1)

[105]: [x, 2x, 3x, 4x, 5x]

[106]: # массив сформирован элементов, если индекс не делится на 5 или 4:
m_2 = [i^2 for i=1:10 if (i%5)!=0 && i%4!=0]
println(m_2)

[107]: [4, 9, 16, 25]
```

Рис. 4: Примеры с массивами.

# Задание для самостоятельной работы 1

Выполнила первое задание для самостоятельной работы:

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

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

```
[176]: union(intersect(A, B), intersect(A, C), intersect(B, C))
```

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

Рис. 5: Задание 1.

## Задание для самостоятельной работы 2

Выполнила второе задание для самостоятельной работы:

2. Привести свои примеры вычисления операций над множествами элементов разных типов.

```
[179]: a = list(range(10))
      b = list(range(5))
      c = list(range(10))

[180]: set(a) & set(b)
Out[180]: {0, 1, 2, 3, 4}

[181]: set(a) & set(c)
Out[181]: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

[182]: set(a) & set(b)
Out[182]: {0, 1, 2, 3, 4}

[183]: set(a) & set(c)
Out[183]: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

[184]: set(a) & set(b)
Out[184]: {0, 1, 2, 3, 4}

[185]: set(a) & set(c)
Out[185]: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

[186]: set(a) & set(b)
Out[186]: {0, 1, 2, 3, 4}

[187]: set(a) & set(c)
Out[187]: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
```

Рис. 6: Задание 2.

### Задание для самостоятельной работы 3

Выполнила третье задание для самостоятельной работы (пункты 1-8):

```

1010: 5. C#değer parametre-sonlandırma anahtarı.
[001]: 0001 [1] Row 1 [0-1:00]
print(001)
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 544, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572, 573, 574, 575, 576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606, 607, 608, 609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799, 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 8
```

Рис. 7: Задание 3 (1-8).

### Задание для самостоятельной работы 3

Выполнила третье задание для самостоятельной работы (пункты 9-11):

[illegible]

Рис. 8: Задание 3 (9-11).

## Задание для самостоятельной работы 3

Выполнила третье задание для самостоятельной работы (пункты 12-13):

```
[236]: A312 = []

for i in 1:25
    push!(A312, {2^i}/i)
end

print(A312)

Any[2.0, 2.0, 2.0866060606060605, 4.0, 6.4, 10.666666666666667, 18.285714285714285, 32.0, 56.800000000000004, 102.4, 186.18181818181818, 341.3333333333333,
3, 430.153661538462, 1370.2857142857142, 2184.5333333333333, 4086.4, 7710.137647058823, 14543.555555555555, 27994.185263157893, 50426.8, 99064.388952380
95, 190450.18181818182, 364722.0869565217, 690050.6666666666, 1.34217728e6]

[238]: A313 = []

for i in 1:30
    push!(A313, "fn$i")
end

print(A313)

Any["fn1", "fn2", "fn3", "fn4", "fn5", "fn6", "fn7", "fn8", "fn9", "fn10", "fn11", "fn12", "fn13", "fn14", "fn15", "fn16", "fn17", "fn18", "fn19", "fn2
0", "fn21", "fn22", "fn23", "fn24", "fn25", "fn26", "fn27", "fn28", "fn29", "fn30"]
```

Рис. 9: Задание 3 (12-13).



## Задание для самостоятельной работы 3

Выполнила третье задание для самостоятельной работы (пункт 14):

```
[14]: n = []
     p = []

     for i in 1:250
         push!(x, rand(0:999))
         push!(y, rand(0:999))
     end

[15]: println(x)

Any{Int64}: 15, 343, 4, 380, 642, 831, 206, 75, 7, 048, 49, 540, 700, 115, 416, 839, 807, 757, 745, 074, 580, 423, 813, 313, 154, 18, 264, 037, 153, 597, 84
4, 87, 508, 253, 557, 786, 785, 464, 462, 079, 089, 387, 473, 136, 281, 95, 108, 303, 917, 141, 182, 676, 249, 564, 862, 784, 949, 835, 308, 306, 218, 74
5, 322, 227, 883, 883, 786, 775, 159, 954, 940, 775, 436, 106, 104, 380, 811, 312, 909, 849, 588, 104, 759, 547, 106, 432, 195, 301, 464, 469, 387,
3, 897, 886, 982, 78, 883, 887, 497, 907, 287, 939, 990, 422, 849, 884, 882, 188, 32, 18, 891, 877, 989, 772, 882, 882, 882, 814, 933, 821, 933, 88, 98,
344, 882, 938, 944, 181, 927, 288, 837, 149, 883, 285, 18, 178, 270, 541, 867, 882, 288, 486, 91, 989, 278, 282, 248, 444, 689, 928, 978, 848, 428,
137, 818, 83, 83, 192, 836, 938, 843, 789, 5, 42, 98, 588, 189, 636, 75, 555, 636, 312, 894, 568, 787, 489, 931, 858, 743, 303, 612, 150, 821, 488, 938,
125, 898, 303, 334, 184, 32, 450, 780, 274, 648, 944, 288, 387, 75, 808, 784, 194, 808, 944, 744, 184, 176, 147, 697, 773, 483, 781, 833, 988, 956, 431, 9
85, 585, 876, 877, 759, 965, 462, 883, 135, 304, 554, 715, 637, 683, 904, 538, 518, 31, 858, 414, 047, 583, 340, 204, 208, 388, 638, 933, 786, 511, 811]

[15]: println(y)

Any{Int64}: 585, 848, 318, 582, 488, 185, 878, 694, 334, 656, 328, 787, 805, 327, 187, 269, 368, 321, 108, 648, 793, 674, 959, 482, 86, 19, 787, 489, 116, 1
2, 238, 889, 188, 389, 832, 71, 671, 578, 751, 587, 588, 144, 541, 288, 212, 887, 948, 586, 074, 213, 183, 98, 388, 589, 831, 771, 888, 718, 469, 437, 22
4, 789, 989, 138, 838, 818, 789, 888, 987, 783, 113, 936, 978, 119, 248, 899, 128, 284, 989, 988, 26, 114, 684, 972, 987, 817, 871, 984, 108, 141, 882, 1
72, 484, 879, 938, 888, 888, 947, 186, 689, 898, 128, 188, 182, 878, 878, 938, 878, 878, 988, 821, 212, 281, 88, 833, 295, 789, 188, 188, 889, 833, 288,
228, 711, 438, 778, 684, 688, 248, 5, 138, 643, 878, 12, 187, 884, 989, 989, 989, 184, 989, 538, 333, 451, 893, 832, 878, 988, 988, 884, 773, 971, 2
82, 784, 328, 569, 859, 257, 945, 19, 258, 972, 350, 342, 400, 893, 424, 644, 487, 251, 419, 40, 828, 648, 548, 927, 348, 15, 646, 146, 113, 18, 999, 82
2, 388, 885, 531, 142, 428, 171, 86, 9, 425, 883, 637, 786, 531, 311, 261, 633, 149, 814, 895, 899, 918, 472, 486, 536, 439, 469, 189, 188, 479, 483, 89
8, 304, 462, 987, 888, 348, 818, 872, 362, 987, 987, 383, 518, 887, 354, 931, 883, 178, 171, 57, 838, 318, 279, 888, 184, 174, 417, 711, 113, 784, 738, 4
2, 843, 681]
```

Рис. 10: Задание 3 (14).

### Задание для самостоятельной работы 3

Выполнила третье задание для самостоятельной работы (пункты 14.1-2):

[illegible]

Рис. 11: Задание 3 (14.1-2).

### Задание для самостоятельной работы 3

Выполнила третье задание для самостоятельной работы (пункт 14.3):

[illegible]

Рис. 12: Задание 3 (14.3).

## Задание для самостоятельной работы 3

Выполнила третье задание для самостоятельной работы (пункты 14.4-5):

```
[206]: res = Base.sum([(exp(-x[i-1]))/(x[i] + 10) for i in 1:249])
[206]: 0.00020522102255307825

[270]: b5 = []

for i in y
    if i > 600
        push!(b5, i)
    end
end

print(b5, '\n', findall(y .> 600))

Any{Int64}[947, 848, 828, 694, 656, 787, 805, 649, 793, 674, 919, 787, 884, 632, 671, 753, 744, 697, 946, 974, 831, 771, 868, 718, 705, 906, 856, 618, 709, 866,
947, 763, 970, 699, 906, 967, 982, 694, 875, 888, 986, 947, 956, 674, 930, 656, 875, 968, 631, 769, 905, 951, 711, 759, 684, 688, 843, 870, 884, 895, 66
3, 926, 895, 852, 972, 894, 773, 971, 704, 928, 859, 945, 972, 693, 844, 828, 846, 927, 846, 999, 822, 809, 865, 837, 786, 835, 914, 895, 999, 918, 839,
899, 888, 628, 972, 947, 987, 657, 911, 883, 926, 868, 715, 784, 738]
[1, 2, 8, 9, 11, 13, 14, 21, 22, 23, 24, 28, 33, 36, 38, 40, 43, 47, 48, 50, 56, 57, 58, 59, 63, 64, 66, 67, 68, 69, 70, 71, 74, 77, 81, 86, 92, 94, 95,
97, 98, 99, 102, 107, 108, 109, 110, 111, 116, 118, 121, 122, 123, 127, 128, 129, 133, 134, 137, 139, 140, 144, 147, 148, 149, 152, 155, 156, 157, 1
59, 161, 164, 168, 170, 175, 176, 178, 181, 185, 186, 188, 196, 197, 198, 202, 204, 205, 206, 207, 211, 217, 221, 223, 224, 226, 227, 230, 232, 233, 237,
240, 244, 246, 247]
```

Рис. 13: Задание 3 (14.4-5).

## Задание для самостоятельной работы 3

Выполнила третье задание для самостоятельной работы (пункт 14.6):

```
[272]: indexes_y = findall(y .> 600)
      b6 = []

      for i in indexes_y
        push!(b6, x[i])
      end

      print(b6)

Any[81, 343, 286, 73, 848, 549, 700, 974, 506, 423, 811, 264, 87, 557, 785, 462, 387, 95, 188, 917, 862, 794, 949, 835, 745, 322, 883, 691, 786, 772, 15
9, 954, 436, 303, 884, 356, 459, 3, 957, 592, 35, 483, 507, 689, 864, 461, 106, 52, 772, 642, 915, 611, 55, 652, 935, 964, 857, 140, 34, 239, 170, 266, 9
49, 278, 202, 639, 574, 975, 429, 137, 83, 592, 841, 90, 339, 112, 896, 787, 650, 376, 821, 330, 730, 174, 640, 75, 764, 554, 800, 864, 147, 990, 505, 87
7, 759, 462, 803, 556, 617, 693, 31, 847, 208, 818, 933]
```

Рис. 14: Задание 3 (14.6).

# Задание для самостоятельной работы 3

## Выполнила третье задание для самостоятельной работы (пункт 14.7):

```
[114] %sum = sum(x) / length(x)
for i = 1
    for j = 0 to 100
        sum = sum(x) / length(x)
    end
end
```

Рис. 15: Задание 3 (14.7).

## Задание для самостоятельной работы 3

Выполнила третье задание для самостоятельной работы (пункт 14.8):

```
[278]: y_max = maximum(y)
count = 0

for i in y
    if abs(i - y_max) <= 200
        count += 1
    end
end

print(count)

61
```

Рис. 16: Задание 3 (14.8).

## Задание для самостоятельной работы 3

Выполнила третье задание для самостоятельной работы (пункты 14.9-10):

```
[282]: count_even = 0

for i in x:
    if i % 2 == 0:
        count_even += 1
    end
end

print("Чётных элементов ", count_even, ", нечётных элементов ", 250 - count_even)
Чётных элементов 125, нечётных элементов 125

[284]: count_seven = 0

for i in x:
    if i % 7 == 0:
        count_seven += 1
    end
end

print(count_seven, " элементов кратно семи")
39 элементов кратно семи
```

Рис. 17: Задание 3 (14.9-10).



## Задание для самостоятельной работы 3

Выполнила третье задание для самостоятельной работы (пункты 14.11-13):

```
[106]: indexes_y = sortperm(y)
       x_sorted_by_y = x[indexes_y]

       print(x_sorted_by_y)

Any[527, 92, 597, 883, 931, 788, 612, 18, 830, 650, 518, 700, 305, 790, 204, 154, 870, 950, 453, 833, 153, 154, 365, 38, 720, 898, 743, 870, 588, 282, 53
9, 334, 539, 367, 984, 483, 377, 235, 418, 414, 388, 593, 745, 575, 255, 141, 757, 228, 86, 382, 641, 75, 217, 334, 5, 880, 844, 83, 681, 838, 381, 773,
414, 840, 152, 915, 807, 736, 482, 950, 343, 353, 249, 4, 715, 388, 7, 40, 115, 611, 464, 512, 468, 887, 287, 547, 505, 340, 408, 703, 936, 903, 42, 477,
284, 555, 108, 476, 451, 344, 283, 55, 738, 386, 811, 380, 639, 748, 59, 642, 823, 658, 637, 907, 113, 186, 340, 300, 879, 650, 256, 942, 432, 247, 484,
264, 10, 576, 115, 846, 771, 560, 864, 985, 988, 610, 422, 464, 336, 473, 382, 15, 566, 641, 381, 935, 691, 821, 877, 772, 557, 75, 174, 147, 896, 974, 8
48, 461, 356, 170, 785, 423, 689, 964, 90, 73, 3, 554, 95, 383, 745, 786, 95, 208, 835, 833, 387, 462, 652, 954, 642, 794, 174, 429, 648, 540, 264, 586,
818, 34, 700, 150, 352, 385, 847, 112, 862, 857, 339, 650, 343, 278, 883, 83, 790, 772, 840, 149, 957, 180, 693, 87, 659, 239, 949, 990, 459, 322, 884, 8
17, 764, 864, 811, 246, 31, 787, 137, 864, 592, 188, 81, 159, 483, 462, 611, 507, 354, 52, 436, 282, 975, 841, 759, 917, 186, 35, 683, 915, 376, 880]

[108]: b10 = reverse(last(sort(x), 10))

       print(b10)

Any[990, 988, 985, 975, 974, 967, 964, 957, 954, 950]

[109]: b11 = unique(x)

       print(b11)

Any[81, 15, 343, 4, 100, 642, 823, 286, 75, 7, 848, 49, 549, 780, 115, 410, 839, 887, 757, 745, 974, 586, 423, 812, 313, 154, 18, 264, 637, 153, 597, 84
4, 87, 588, 353, 557, 798, 785, 464, 462, 679, 988, 387, 473, 730, 182, 95, 188, 381, 937, 141, 262, 870, 249, 946, 862, 794, 949, 835, 390, 306, 228, 32
2, 217, 883, 691, 786, 772, 159, 954, 942, 771, 436, 10, 334, 383, 611, 532, 650, 884, 780, 730, 547, 356, 432, 595, 382, 484, 720, 450, 367, 3, 957, 86
4, 352, 35, 483, 897, 477, 507, 247, 539, 422, 689, 461, 180, 52, 641, 377, 385, 462, 433, 414, 935, 86, 55, 344, 652, 939, 964, 181, 527, 256, 857, 149,
889, 255, 34, 575, 239, 170, 967, 681, 266, 278, 282, 840, 639, 574, 975, 889, 429, 137, 638, 91, 838, 336, 841, 789, 5, 412, 90, 368, 339, 638, 75, 555,
656, 112, 896, 560, 787, 469, 931, 650, 743, 365, 612, 376, 821, 480, 130, 125, 838, 283, 284, 92, 476, 174, 648, 866, 267, 764, 554, 880, 746, 184, 576,
147, 659, 775, 59, 825, 990, 936, 451, 983, 585, 877, 795, 985, 235, 550, 715, 617, 693, 904, 518, 31, 658, 847, 593, 540, 280, 816, 925, 512]
```

Рис. 18: Задание 3 (14.11-13).

## Задание для самостоятельной работы 4

Выполнила четвёртое задание для самостоятельной работы:

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

```
[292]: squares = [i**2 for i in range(1,101)]
```

```
print(squares)
```

```
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 289, 324, 361, 400, 441, 484, 529, 576, 625, 676, 729, 784, 841, 900, 961, 1024, 1089, 1156, 1225, 1296, 1369, 1444, 1521, 1600, 1681, 1764, 1849, 1936, 2025, 2116, 2209, 2304, 2401, 2500, 2601, 2704, 2809, 2916, 3025, 3136, 3249, 3364, 3481, 3600, 3721, 3844, 3969, 4096, 4225, 4356, 4489, 4624, 4761, 4900, 5041, 5184, 5329, 5476, 5625, 5776, 5929, 6084, 6241, 6400, 6561, 6724, 6889, 7056, 7225, 7396, 7569, 7744, 7921, 8100, 8281, 8464, 8649, 8836, 9025, 9216, 9409, 9604, 9801, 10000]
```

Рис. 19: Задание 4.

# Задание для самостоятельной работы 5

Выполнила пятое задание для самостоятельной работы:

5. Подключите пакет Primes (функции для вычисления простых чисел). Сгенерируйте массив myprimes, в котором будут храниться первые 168 простых чисел. Определите 89-е наименьшее простое число. Получите срез массива с 89-го до 99-го элемента включительно, содержащий наименьшие простые числа.

```
[380]: using Primes

myprimes = primes(prime(168))

print(myprimes)

[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239, 241, 251, 257, 263, 269, 271, 277, 281, 283, 293, 307, 311, 313, 317, 331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401, 409, 419, 421, 431, 433, 439, 443, 449, 457, 461, 463, 467, 479, 483, 491, 499, 503, 509, 521, 523, 541, 547, 557, 563, 569, 571, 577, 587, 593, 599, 601, 607, 611, 617, 619, 631, 641, 643, 647, 653, 659, 661, 673, 677, 683, 691, 701, 709, 719, 727, 733, 739, 743, 751, 757, 761, 769, 773, 787, 797, 809, 811, 821, 823, 827, 829, 839, 853, 857, 859, 863, 877, 881, 883, 887, 907, 911, 919, 929, 937, 941, 947, 953, 967, 971, 977, 983, 991, 997]
```

```
[389]: print(myprimes[89])

461
```

```
[386]: print(myprimes[89:99])

[461, 463, 467, 479, 483, 491, 499, 503, 509, 521, 523]
```

Рис. 20: Задание 5.

## Задание для самостоятельной работы 6

Выполнила шестое задание для самостоятельной работы:

```
6. Вычислите выражения.

[101] result = 0
for i in 10:100
    result += 2*i + 4*i^2
end
show(result)
20812735

[102] result = 0
for i in 1:25
    result += (2*i)/3 + (3*i)/(11*i)
end
show(result)
2.1291794508143806e9

[103] result = 1
tap = 1
for i in 2:2:30
    tap *= i/(i+1)
    result += tap
end
show(result)
6.978346537897668
```

Рис. 21: Задание 6.

## Заключение

---

Изучила несколько структур данных, реализованных в Julia, научилась применять их и операции над ними для решения задач.