# Министерство образования и науки Российской Федерации

## Федеральное государственное бюджетное образовательное учреждение высшего образования

## «Новосибирский государственный технический университет»

NSTU_Logo_blue

## Кафедра теоретической и прикладной информатики

### Лабораторная работа № 4 по дисциплине «Компьютерное моделирование»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| сигма градиент синий1 | Факультет: | ПМИ |  |  |
| Группа: | ПМИ-62 |  |  |
| Студенты: | Ершов П.К., Мамонова Е.В., Цыденов З.Б. |  |  |
| Вариант: | 10 |  |  |
| Преподаватель: | Волкова В.М. |  |  |

Новосибирск

2019

1. **Цель работы**

Научиться моделировать значения непрерывно распределённой случайной величины методом обратной функции и проводить статистический анализ сгенерированных данных.

1. **Задание**

| Вариант | Распределение | Параметры закона распределения | Непараметрический критерий |
| --- | --- | --- | --- |
| 10 | Логистическое | μ = 10, ν = 1; μ = –2.2, ν = 2.3; μ = 0, ν = 0.5 | Критерий ω2-Крамера-Мизеса-Смирнова |

1. **Описание формата входного файла**

В файле data.txt находятся входные данные для всех тестов.

41 – коэффициент a

51 – коэффициент b

13 – коэффициент c

1.0001 – число m, остатком от которого является псевдослучайное число

1 – текущее значение xn в алгоритме псевдослучайных чисел из второй лабораторной

50 – максимальное количество чисел N в последовательности

200 – максимальное количество чисел N в последовательности

1000 - максимальное количество чисел N в последовательности

10 – μ параметр распределения

-2.2 – μ параметр распределения

0 – μ параметр распределения

1 – v параметр распределения

2.3 – v параметр распределения

0.5 – v параметр распределения

0.05 – уровень значимости α.

Получение формулы распределения:

Получение:

1. **Ход работы**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Длина последовательности** | 50 | | | | | |
| **Параметры распределения** | μ | v | μ | v | μ | v |
| 10 | 1 | -2.2 | 2.3 | 0 | 0.5 |
| **Последовательность** | 17.542, 13.774, 7.394, 9.562, 9.549, 9.638, 8.735, 11.547, 14.157, 6.582, 5.145, 12.435, 11.694, 7.831, 8.923, 9.439, 10.969, 10.504, 9.319, 10.489, 11.354, 9.436, 13.422, 10.377, 10.945, 11.36, 9.178, 10.552, 11.488, 10.008, 7.544, 11.238, 9.081, 9.694, 11.459, 9.282, 11.795, 9.544, 8.624, 10.566, 14.25, 10.175, 6.203, 8.784, 11.345, 11.933, 12.107, 8.285, 8.564, 11.255 | | 15.147, 6.48, -8.194, -3.207, -3.236, -3.034, -5.109, 1.357, 7.362, -10.062, -13.367, 3.401, 1.696, -7.189, -4.678, -3.49, 0.028, -1.04, -3.767, -1.076, 0.914, -3.497, 5.671, -1.332, -0.026, 0.927, -4.091, -0.931, 1.222, -2.182, -7.849, 0.648, -4.313, -2.903, 1.156, -3.851, 1.929, -3.248, -5.365, -0.897, 7.575, -1.797, -10.934, -4.996, 0.895, 2.246, 2.646, -6.145, -5.502, 0.686 | | 3.771, 1.887, -1.303, -0.219, -0.225, -0.181, -0.632, 0.773, 2.079, -1.709, -2.428, 1.218, 0.847, -1.085, -0.539, -0.28, 0.484, 0.252, -0.341, 0.244, 0.677, -0.282, 1.711, 0.189, 0.473, 0.68, -0.411, 0.276, 0.744, 0.004, -1.228, 0.619, -0.459, -0.153, 0.73, -0.359, 0.898, -0.228, -0.688, 0.283, 2.125, 0.088, -1.899, -0.608, 0.673, 0.967, 1.053, -0.858, -0.718, 0.627 | |
| **Время моделирования** | 4.447e-05 | | 8.273e-05 | | 0.000133 | |
| **Относительные частоты** | 0.06, 0.06, 0.34, 0.22, 0.22, 0.08, 0.0, 0.02 | | 0.06, 0.06, 0.34, 0.22, 0.22, 0.08, 0.0, 0.02 | | 0.06, 0.06, 0.34, 0.22, 0.22, 0.08, 0.0, 0.0 | |
| **Значение статистики S** | 17.608 | | 17.608 | | 9.411 | |
| **Достигнутый уровень значимости** | 0.014 | | 0.014 | | 0.224 | |
| **Результаты проверки критерия *χ*2** | Не пройден | | Не пройден | | Пройден | |
| **Значение статистики S** | 0.134 | | 0.134 | | 0.134 | |
| **Достигнутый уровень значимости** | 0.445 | | 0.445 | | 0.445 | |
| **Результаты проверки критерия** **ω2-Крамера-Мизеса-Смирнова** | Пройден | | Пройден | | Пройден | |
| **Вывод о качестве сгенерированной последовательности** | Является не очень качественной, так как не равномерная | | Является не очень качественной, так как не равномерная | | Можно считать качественной, так как, возможно, равномерная | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Длина последовательности** | 200 | | | | | |
| **Параметры распределения** | μ | v | μ | v | μ | v |
| 10 | 1 | -2.2 | 2.3 | 0 | 0.5 |
| **Последовательность** | 17.542, 13.774, 7.394, 9.562, 9.549, 9.638, 8.735, 11.547, 14.157, 6.582, 5.145, 12.435, 11.694, 7.831, 8.923, 9.439, 10.969, 10.504, 9.319, 10.489, 11.354, 9.436, 13.422, 10.377, 10.945, 11.36, 9.178, 10.552, 11.488, 10.008, 7.544, 11.238, 9.081, 9.694, 11.459, 9.282, 11.795, 9.544, 8.624, 10.566, 14.25, 10.175, 6.203, 8.784, 11.345, 11.933, 12.107, 8.285, 8.564, 11.255, 10.225, 10.312, 8.837, 11.018, 10.591, 9.338, 8.482, 11.636, 10.996, 11.678, 9.686, 9.222, 12.45, 10.946, 6.373, 11.098, 8.812, 11.561, 9.271, 7.012, 8.278, 8.085, 8.22, 9.402, 9.097, 10.266, 10.782, 12.005, 8.4, 10.538, 7.454, 13.597, 9.087, 8.821, 7.636, 15.112, 9.542, 11.219, 5.315, 6.847, 11.762, 11.216, 11.654, 7.42, 11.643, 10.807, 9.022, 9.366, 9.595, 14.652, 8.707, 10.603, 10.369, 10.066, 9.637, 10.562, 12.283, 9.176, 11.072, 9.322, 13.037, 10.071, 9.425, 10.409, 9.895, 11.673, 10.085, 9.464, 6.826, 10.667, 11.793, 10.873, 8.672, 14.809, 10.612, 9.734, 7.869, 10.381, 9.504, 11.42, 8.662, 11.511, 10.144, 10.135, 7.363, 11.999, 9.49, 7.955, 11.548, 14.179, 7.022, 10.839, 10.176, 10.272, 10.44, 10.967, 8.979, 10.523, 11.229, 10.624, 9.573, 10.24, 9.784, 12.605, 10.654, 10.69, 11.419, 10.698, 10.75, 10.15, 12.762, 11.241, 10.74, 9.986, 8.75, 10.255, 9.287, 8.563, 8.041, 10.928, 10.982, 9.778, 7.235, 11.978, 8.2, 10.025, 9.798, 7.15, 11.301, 8.976, 11.174, 9.337, 9.825, 8.417, 10.633, 5.913, 7.657, 12.254, 11.197, 10.749, 9.694, 9.105, 13.087, 10.963, 11.677, 9.12, 6.4, 10.662, 10.213, 8.651 | | 15.147, 6.48, -8.194, -3.207, -3.236, -3.034, -5.109, 1.357, 7.362, -10.062, -13.367, 3.401, 1.696, -7.189, -4.678, -3.49, 0.028, -1.04, -3.767, -1.076, 0.914, -3.497, 5.671, -1.332, -0.026, 0.927, -4.091, -0.931, 1.222, -2.182, -7.849, 0.648, -4.313, -2.903, 1.156, -3.851, 1.929, -3.248, -5.365, -0.897, 7.575, -1.797, -10.934, -4.996, 0.895, 2.246, 2.646, -6.145, -5.502, 0.686, -1.683, -1.482, -4.875, 0.142, -0.842, -3.722, -5.691, 1.562, 0.091, 1.659, -2.923, -3.99, 3.434, -0.023, -10.541, 0.324, -4.932, 1.39, -3.876, -9.072, -6.16, -6.605, -6.294, -3.575, -4.277, -1.589, -0.402, 2.411, -5.88, -0.963, -8.057, 6.073, -4.3, -4.912, -7.637, 9.558, -3.253, 0.604, -12.975, -9.452, 1.853, 0.596, 1.604, -8.133, 1.578, -0.343, -4.449, -3.659, -3.131, 8.5, -5.175, -0.813, -1.352, -2.047, -3.034, -0.908, 3.05, -4.095, 0.265, -3.76, 4.785, -2.036, -3.523, -1.26, -2.441, 1.647, -2.004, -3.432, -9.5, -0.666, 1.924, -0.192, -5.254, 8.861, -0.791, -2.813, -7.1, -1.323, -3.341, 1.066, -5.278, 1.276, -1.87, -1.888, -8.265, 2.397, -3.373, -6.904, 1.36, 7.411, -9.05, -0.271, -1.794, -1.574, -1.188, 0.025, -4.547, -0.997, 0.628, -0.765, -3.182, -1.649, -2.696, 3.792, -0.696, -0.613, 1.064, -0.594, -0.474, -1.855, 4.153, 0.655, -0.498, -2.231, -5.076, -1.614, -3.839, -5.505, -6.705, -0.067, 0.059, -2.71, -8.561, 2.349, -6.34, -2.143, -2.665, -8.755, 0.792, -4.556, 0.501, -3.725, -2.603, -5.842, -0.744, -11.599, -7.59, 2.984, 0.552, -0.478, -2.903, -4.257, 4.901, 0.015, 1.657, -4.224, -10.481, -0.677, -1.711, -5.304 | | 3.771, 1.887, -1.303, -0.219, -0.225, -0.181, -0.632, 0.773, 2.079, -1.709, -2.428, 1.218, 0.847, -1.085, -0.539, -0.28, 0.484, 0.252, -0.341, 0.244, 0.677, -0.282, 1.711, 0.189, 0.473, 0.68, -0.411, 0.276, 0.744, 0.004, -1.228, 0.619, -0.459, -0.153, 0.73, -0.359, 0.898, -0.228, -0.688, 0.283, 2.125, 0.088, -1.899, -0.608, 0.673, 0.967, 1.053, -0.858, -0.718, 0.627, 0.112, 0.156, -0.581, 0.509, 0.295, -0.331, -0.759, 0.818, 0.498, 0.839, -0.157, -0.389, 1.225, 0.473, -1.813, 0.549, -0.594, 0.78, -0.364, -1.494, -0.861, -0.958, -0.89, -0.299, -0.452, 0.133, 0.391, 1.002, -0.8, 0.269, -1.273, 1.798, -0.456, -0.589, -1.182, 2.556, -0.229, 0.61, -2.342, -1.577, 0.881, 0.608, 0.827, -1.29, 0.821, 0.404, -0.489, -0.317, -0.202, 2.326, -0.647, 0.302, 0.184, 0.033, -0.181, 0.281, 1.141, -0.412, 0.536, -0.339, 1.518, 0.036, -0.288, 0.204, -0.052, 0.836, 0.043, -0.268, -1.587, 0.334, 0.897, 0.437, -0.664, 2.405, 0.306, -0.133, -1.065, 0.191, -0.248, 0.71, -0.669, 0.756, 0.072, 0.068, -1.319, 0.999, -0.255, -1.023, 0.774, 2.089, -1.489, 0.419, 0.088, 0.136, 0.22, 0.484, -0.51, 0.261, 0.615, 0.312, -0.213, 0.12, -0.108, 1.303, 0.327, 0.345, 0.71, 0.349, 0.375, 0.075, 1.381, 0.621, 0.37, -0.007, -0.625, 0.127, -0.356, -0.718, -0.979, 0.464, 0.491, -0.111, -1.383, 0.989, -0.9, 0.012, -0.101, -1.425, 0.65, -0.512, 0.587, -0.332, -0.088, -0.792, 0.317, -2.043, -1.172, 1.127, 0.598, 0.374, -0.153, -0.447, 1.544, 0.481, 0.839, -0.44, -1.8, 0.331, 0.106, -0.675 | |
| **Время моделирования** | 0.0002 | | 0.0002 | | 0.0001 | |
| **Относительные частоты** | 0.02, 0.055, 0.085, 0.24, 0.26, 0.225, 0.055, 0.03, 0.025, 0.0, 0.005 | | 0.02, 0.055, 0.085, 0.24, 0.26, 0.225, 0.055, 0.03, 0.025, 0.0, 0.0 | | 0.02, 0.055, 0.085, 0.24, 0.26, 0.225, 0.055, 0.03, 0.025, 0.0, 0.005 | |
| **Значение статистики S** | 13.886 | | 11.357 | | 13.886 | |
| **Достигнутый уровень значимости** | 0.178 | | 0.33 | | 0.178 | |
| **Результаты проверки критерия *χ*2** | Пройден | | Пройден | | Пройден | |
| **Значение статистики S** | 0.146 | | 0.146 | | 0.146 | |
| **Достигнутый уровень значимости** | 0.404 | | 0.404 | | 0.404 | |
| **Результаты проверки критерия** **ω2-Крамера-Мизеса-Смирнова** | Пройден | | Пройден | | Пройден | |
| **Вывод о качестве сгенерированной последовательности** | Можно считать качественной, так как, возможно, равномерная | | Можно считать качественной, так как, возможно, равномерная | | Можно считать качественной, так как, возможно, равномерная | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Длина последовательности** | 1000 | | | | | |
| **Параметры распределения** | μ | v | μ | v | μ | v |
| 10 | 1 | -2.2 | 2.3 | 0 | 0.5 |
| **Время моделирования** | 0.0007 | | 0.0007 | | 0.00068 | |
| **Относительные частоты** | 0.002, 0.0, 0.013, 0.017, 0.057, 0.096, 0.185, 0.237, 0.217, 0.108, 0.039, 0.018, 0.007, 0.003, 0.001 | | 0.002, 0.0, 0.013, 0.017, 0.057, 0.096, 0.185, 0.237, 0.217, 0.108, 0.039, 0.018, 0.007, 0.003, 0.0 | | 0.002, 0.0, 0.013, 0.017, 0.057, 0.096, 0.185, 0.237, 0.217, 0.108, 0.039, 0.018, 0.007, 0.003, 0.001 | |
| **Значение статистики S** | 17.535 | | 18.437 | | 17.535 | |
| **Достигнутый уровень значимости** | 0.228 | | 0.187 | | 0.228 | |
| **Результаты проверки критерия *χ*2** | Пройден | | Пройден | | Пройден | |
| **Значение статистики S** | 0.199 | | 0.199 | | 0.199 | |
| **Достигнутый уровень значимости** | 0.269 | | 0.269 | | 0.269 | |
| **Результаты проверки критерия** **ω2-Крамера-Мизеса-Смирнова** | Пройден | | Пройден | | Пройден | |
| **Вывод о качестве сгенерированной последовательности** | Можно считать качественной, так как, возможно, равномерная | | Можно считать качественной, так как, возможно, равномерная | | Можно считать качественной, так как, возможно, равномерная | |

**Графики.**

График функции распределения

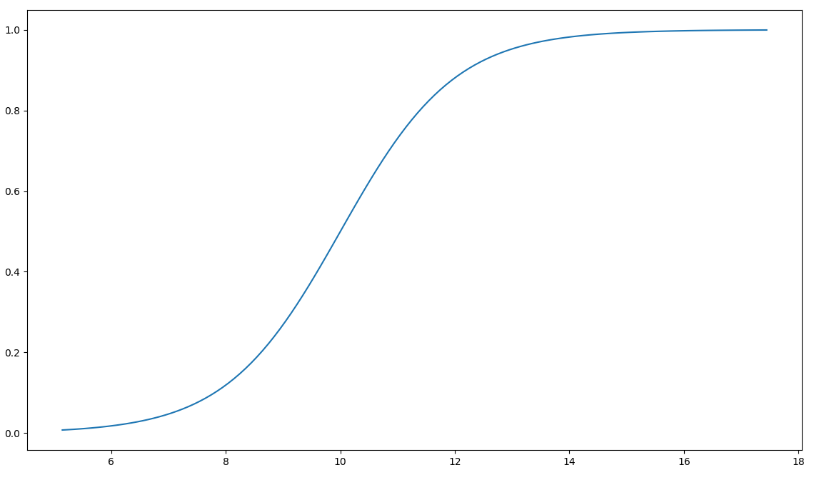
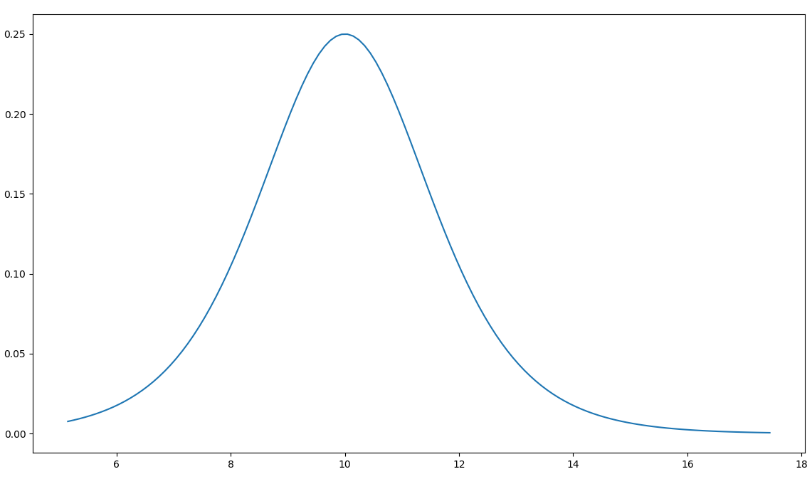


График функции плотности распределения



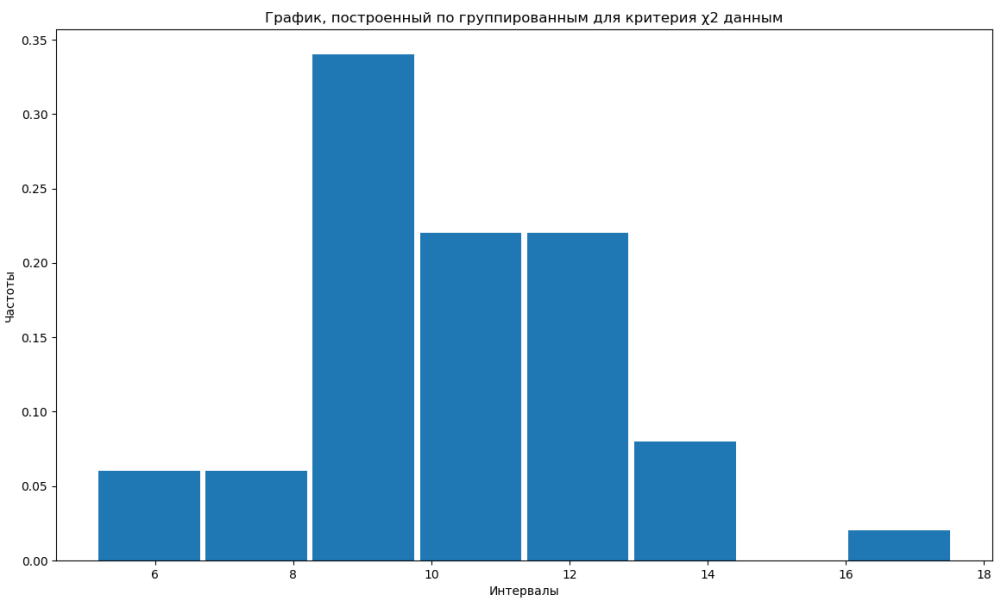


График функции распределения

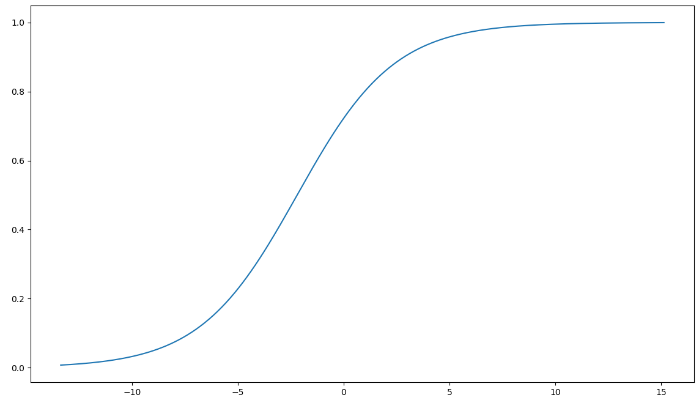
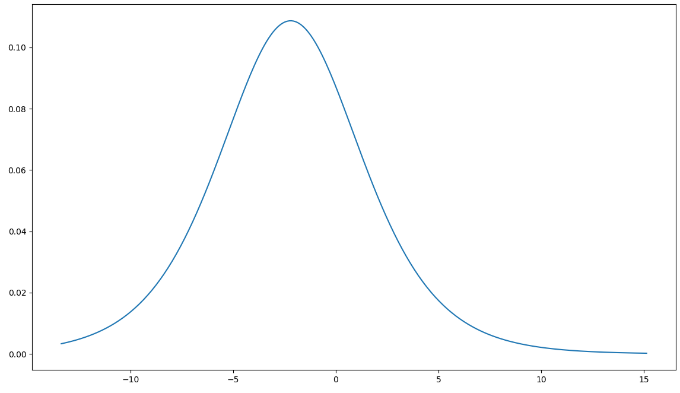


График функции плотности распределения



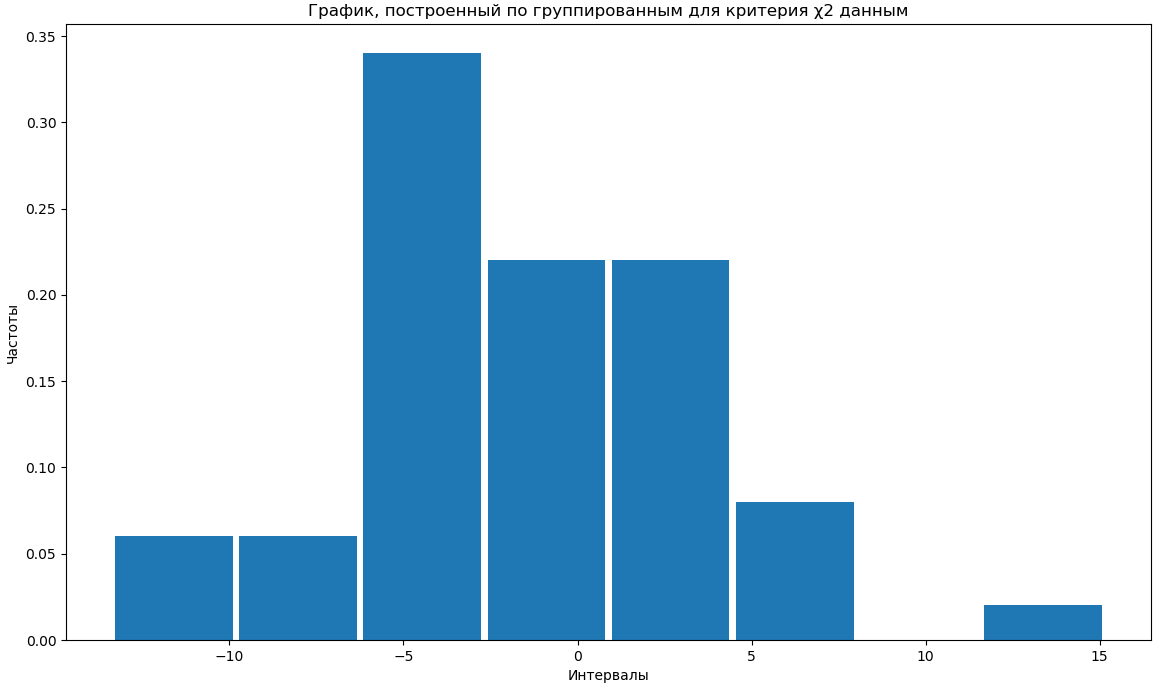


График функции распределения

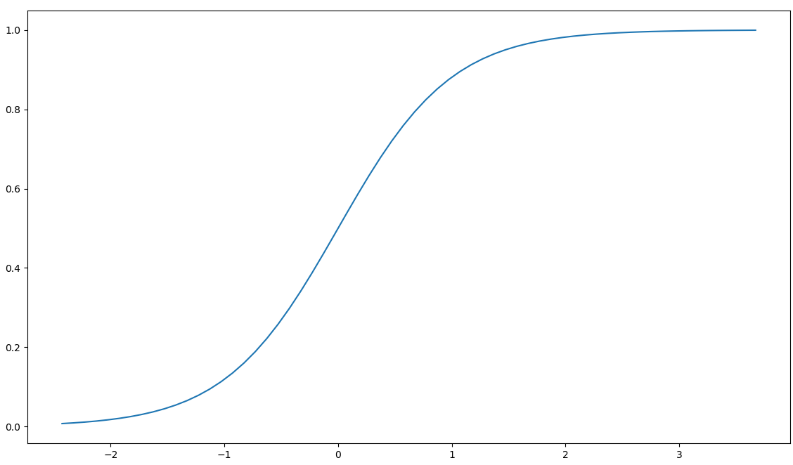
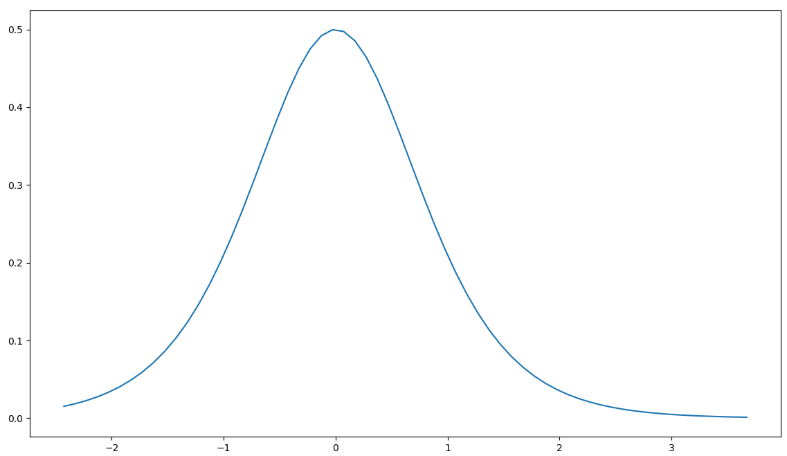


График функции плотности распределения



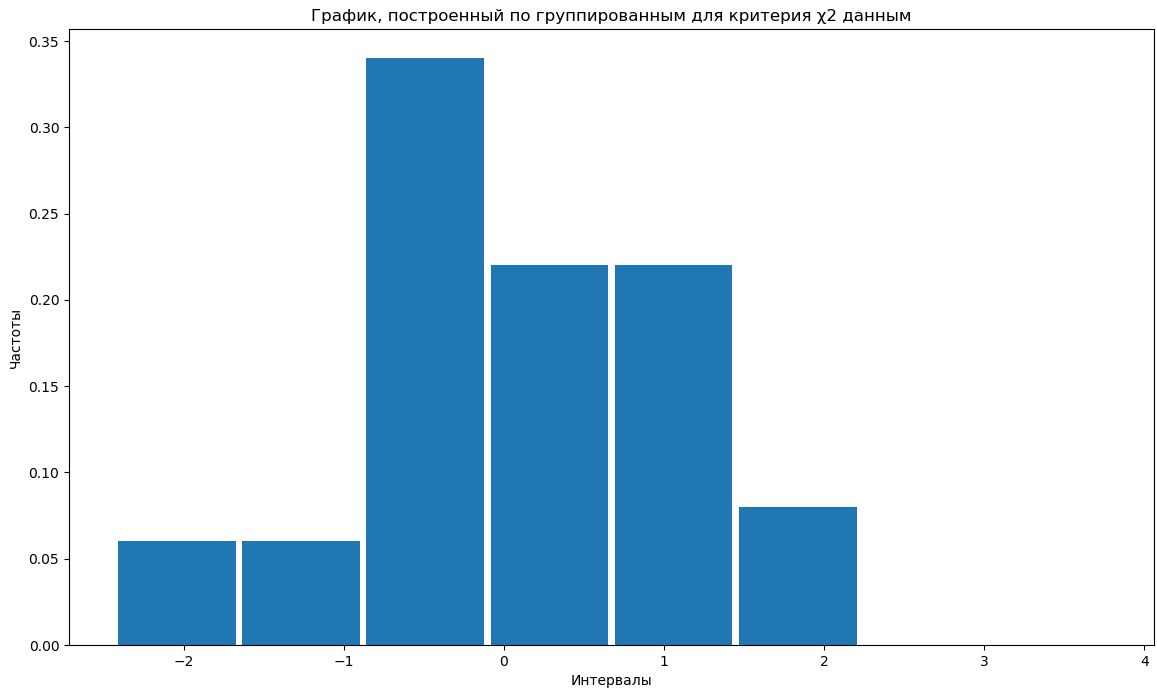


График функции распределения

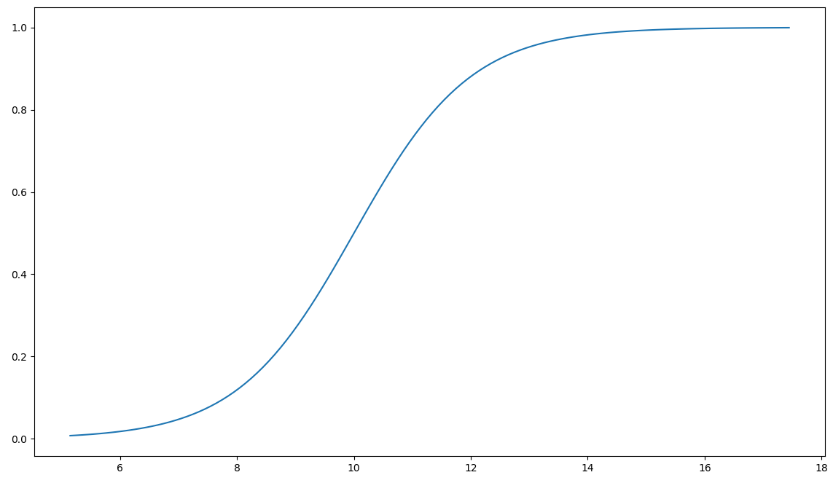
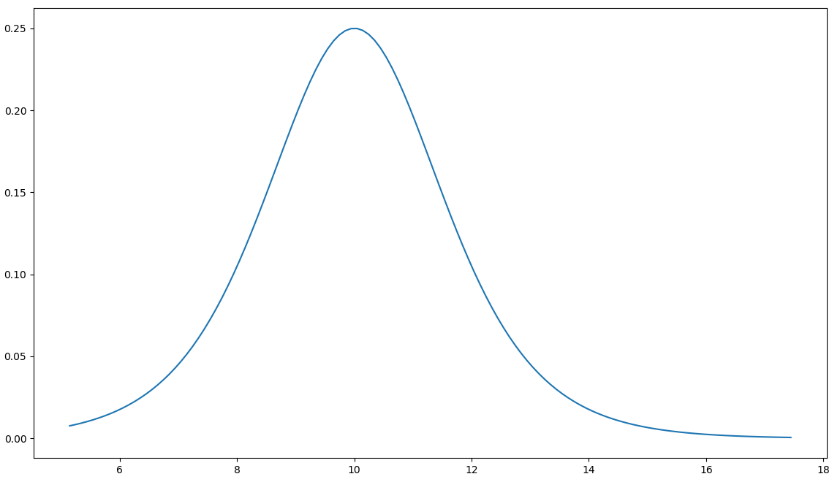


График функции плотности распределения



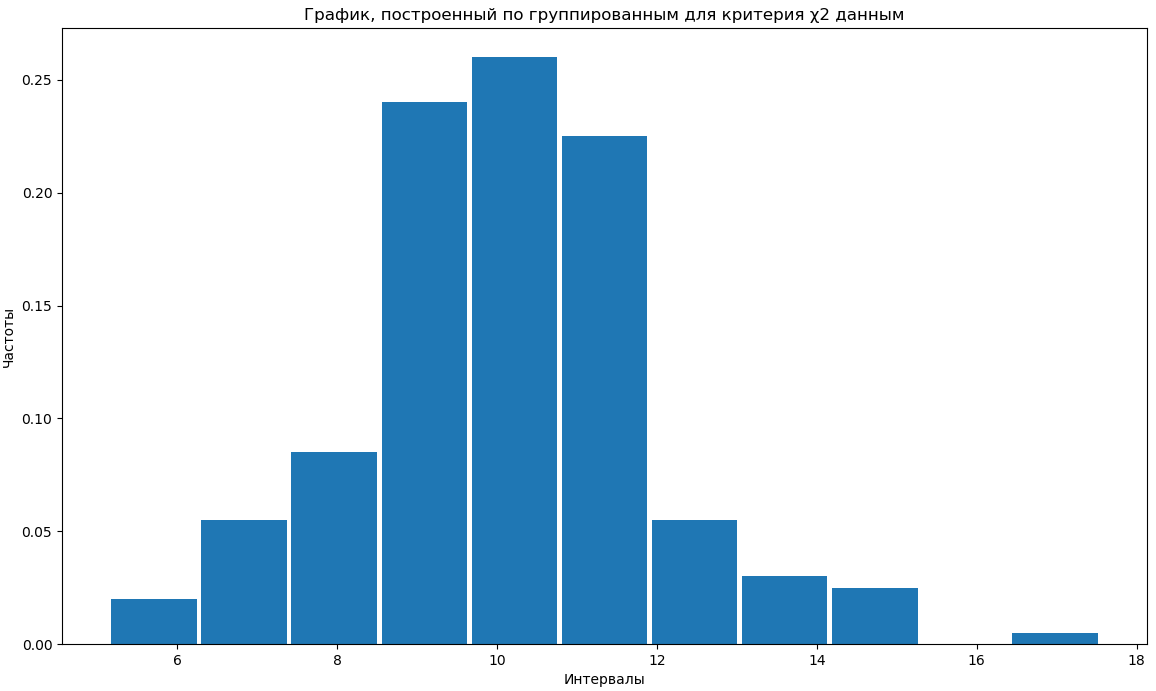


График функции распределения

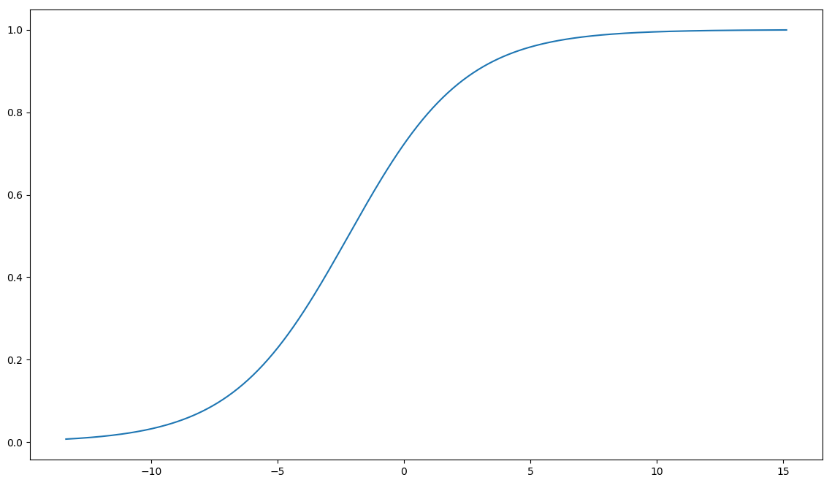
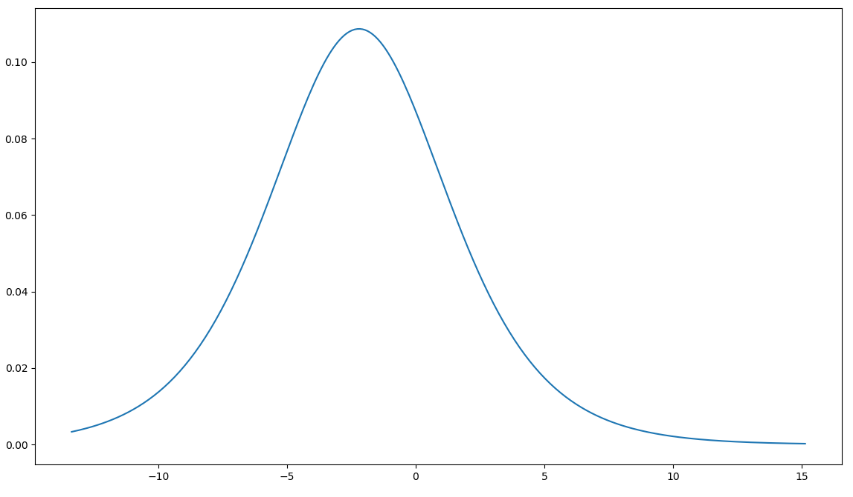


График функции плотности распределения



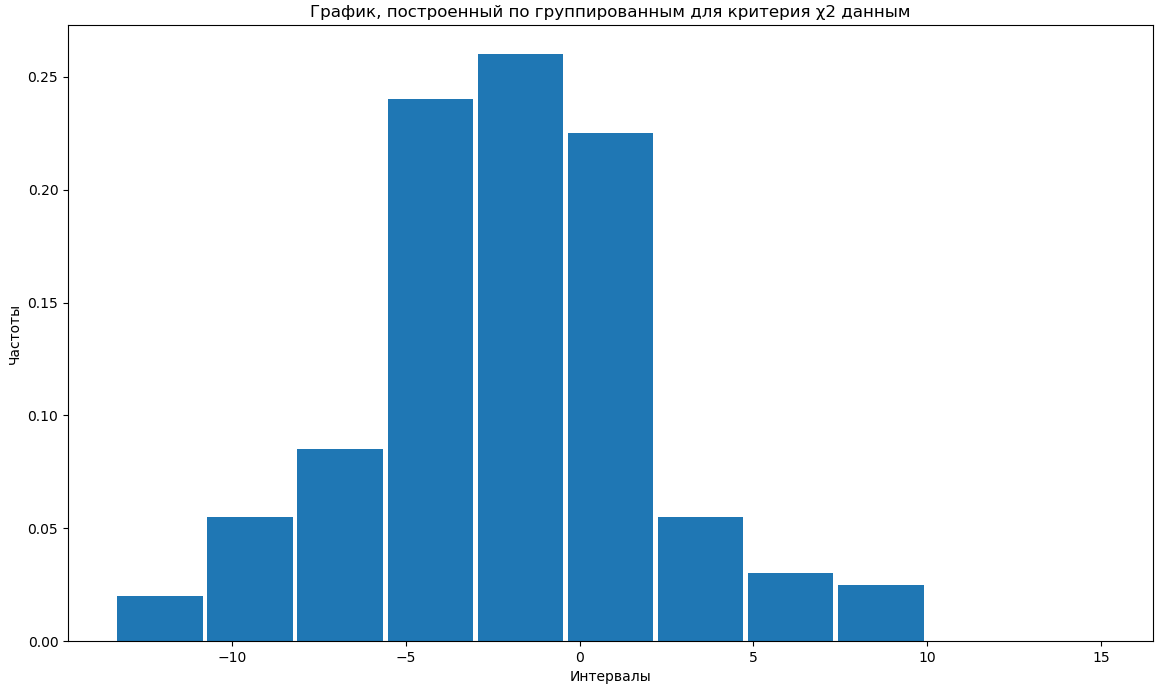


График функции распределения

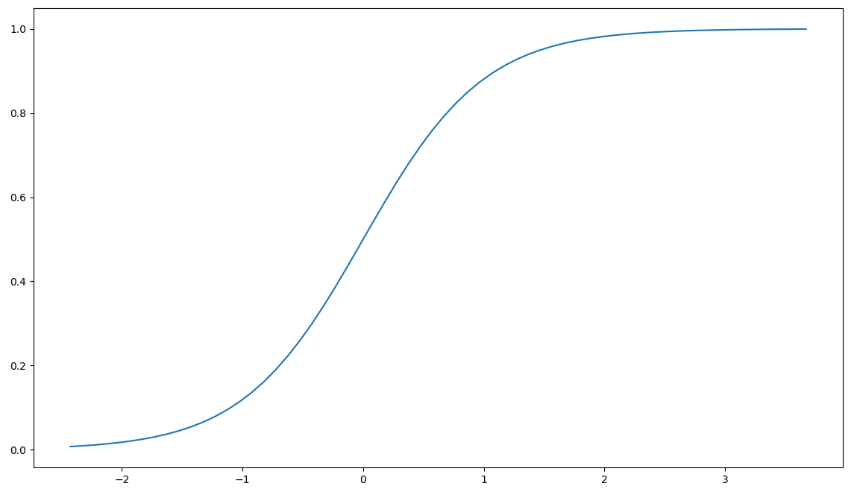


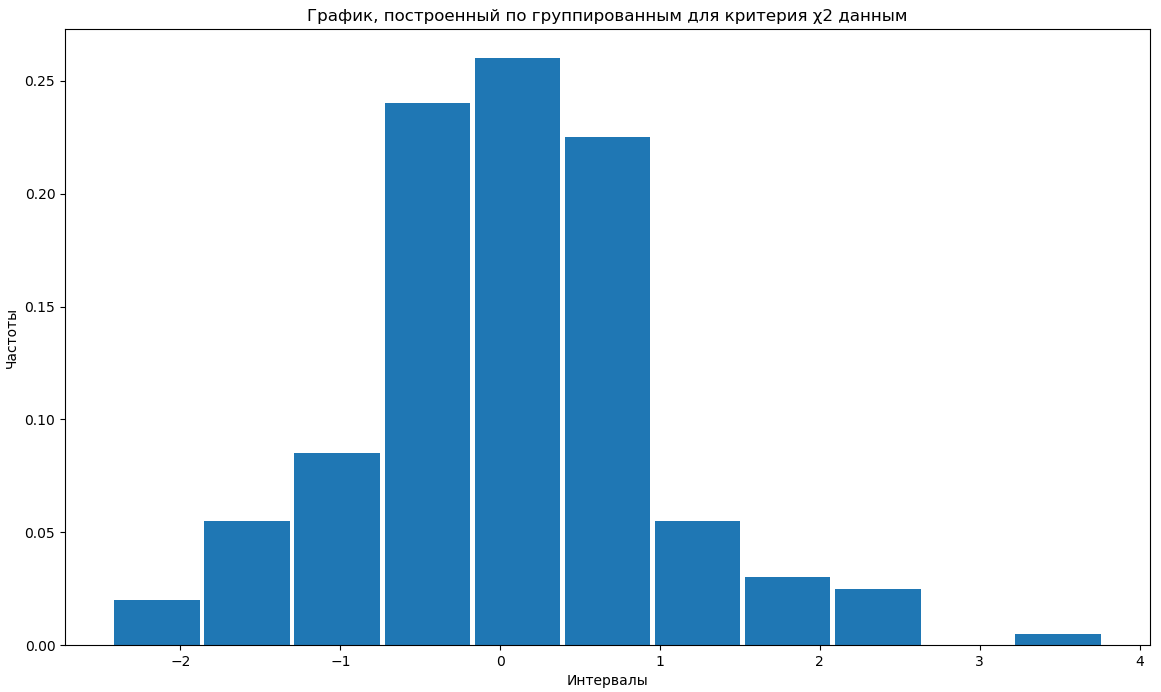
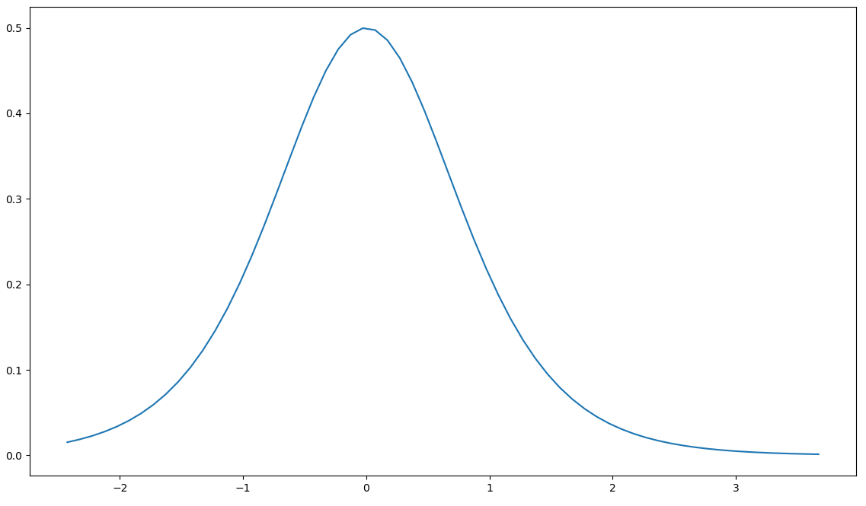
График функции плотности распределения 

График функции распределения

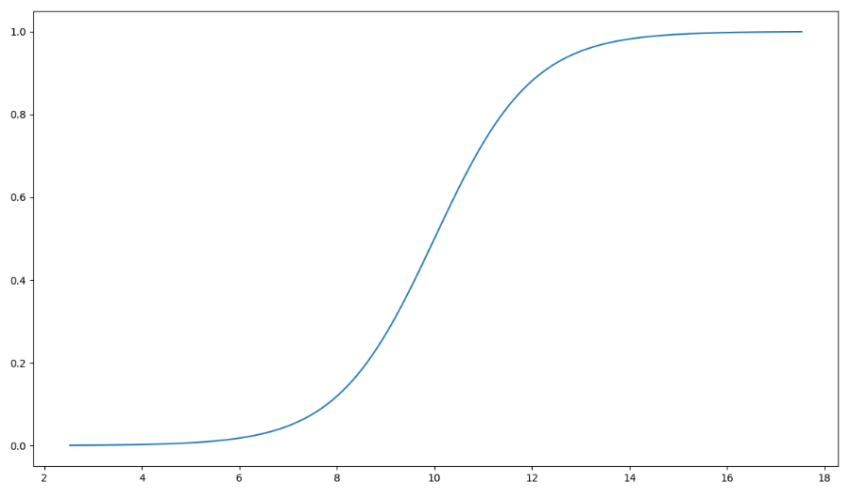
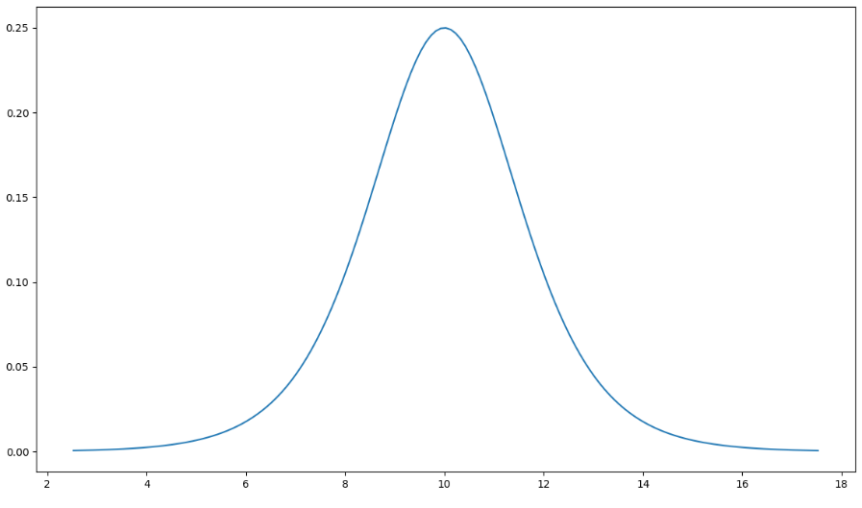


График функции плотности распределения 

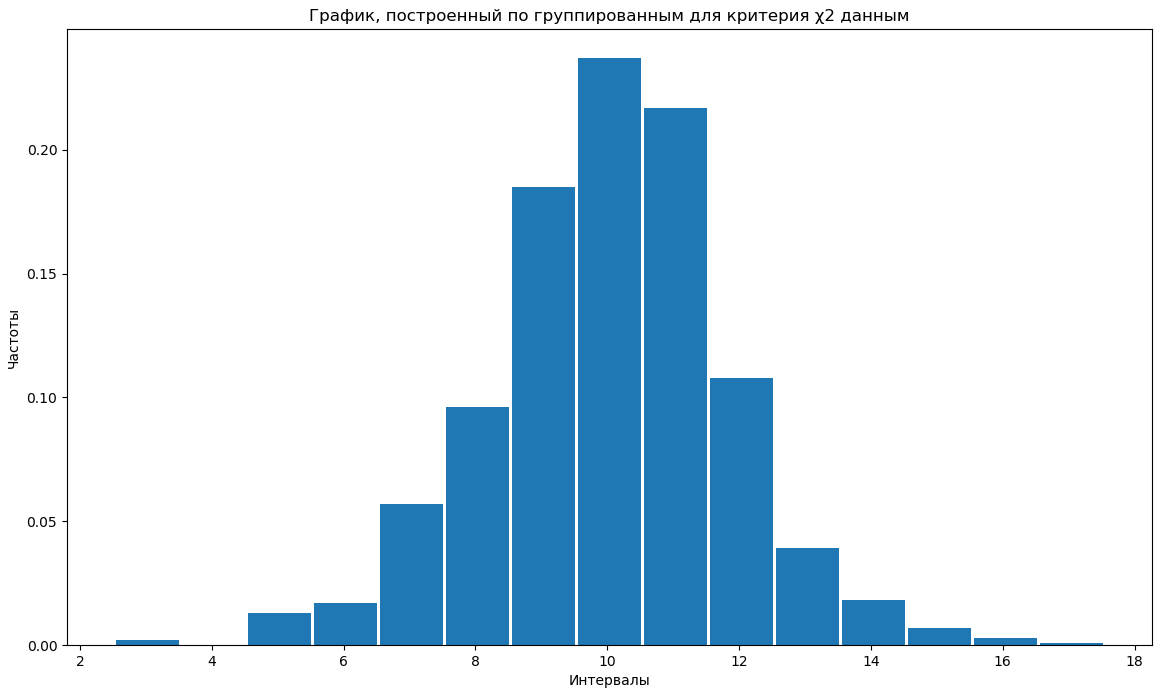


График функции распределения

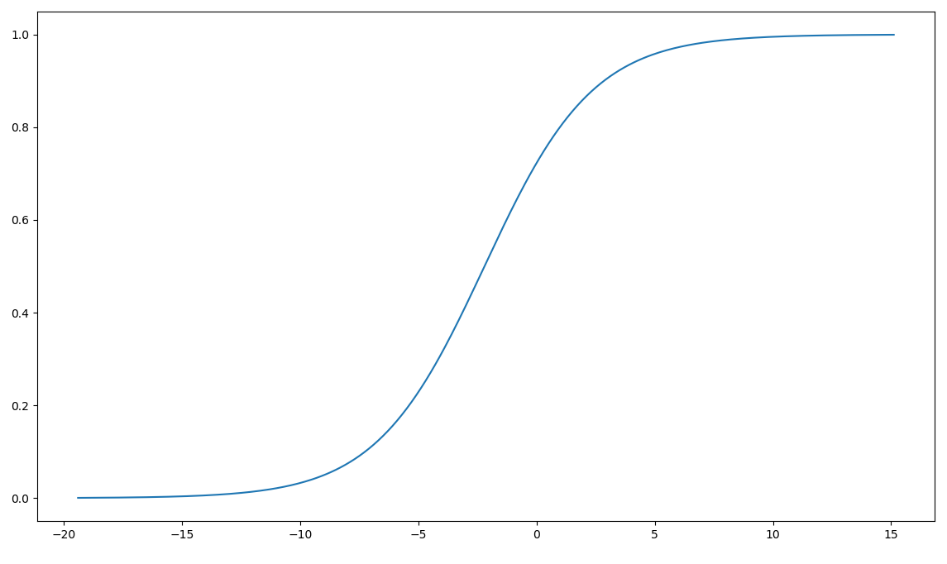


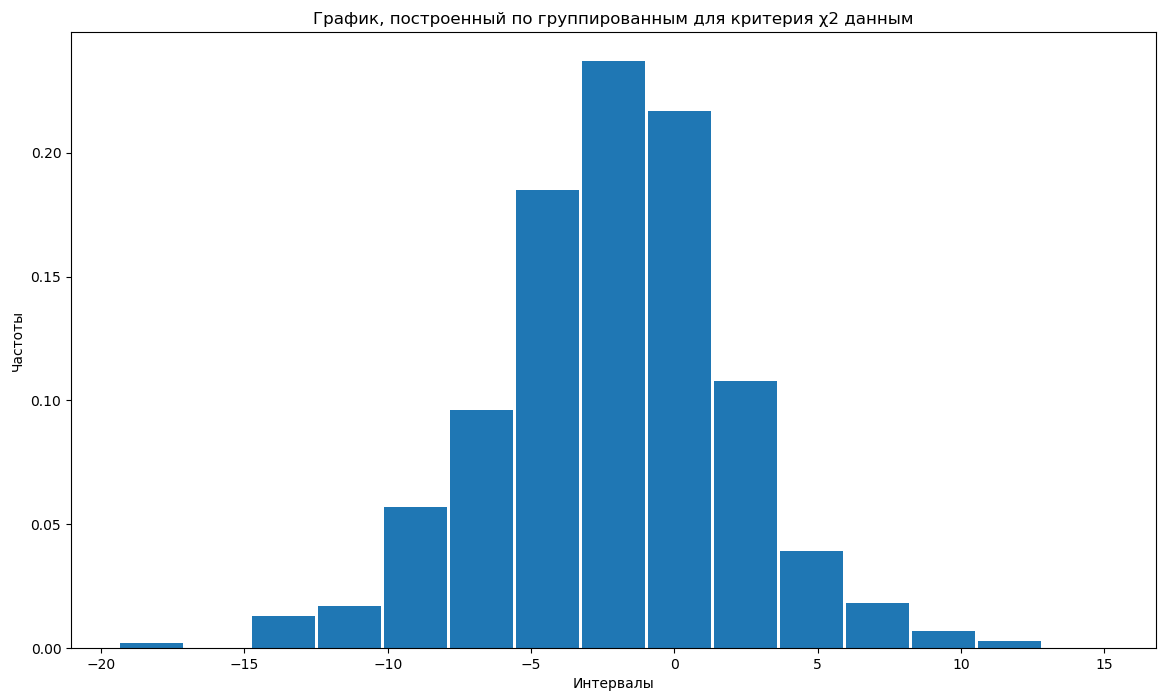
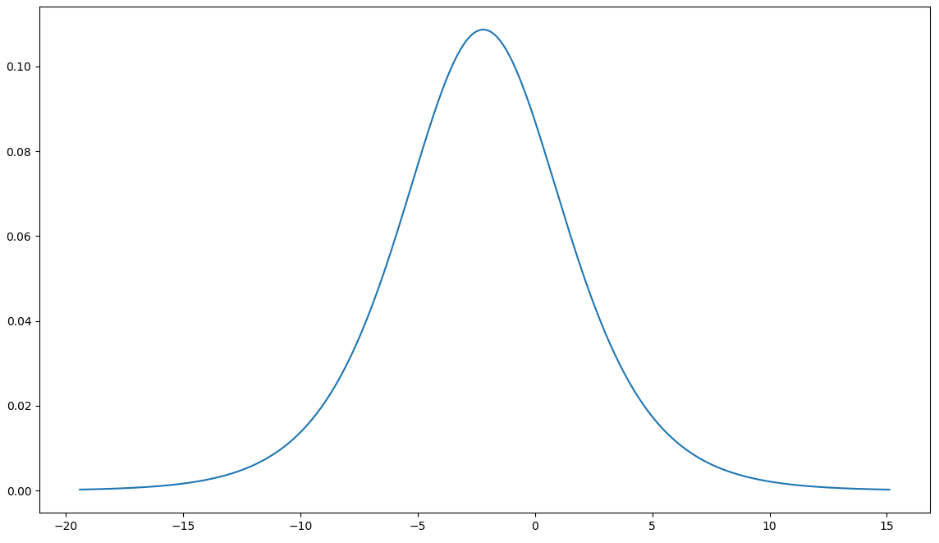
График функции плотности распределения 

График функции распределения

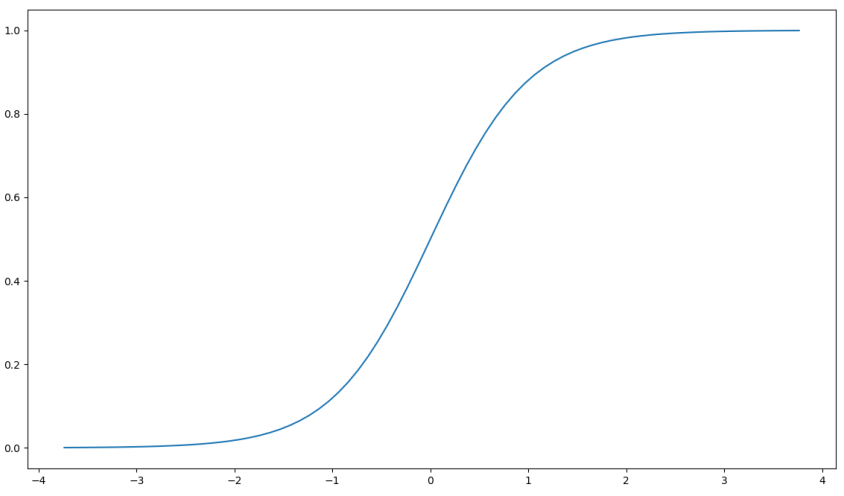
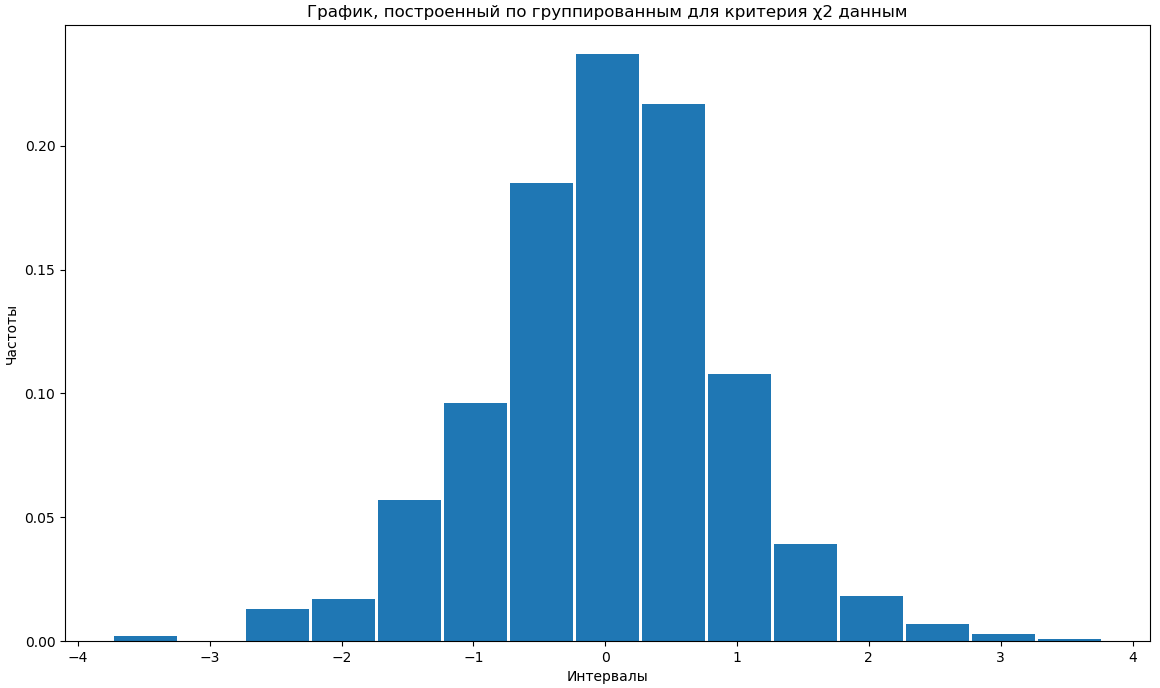
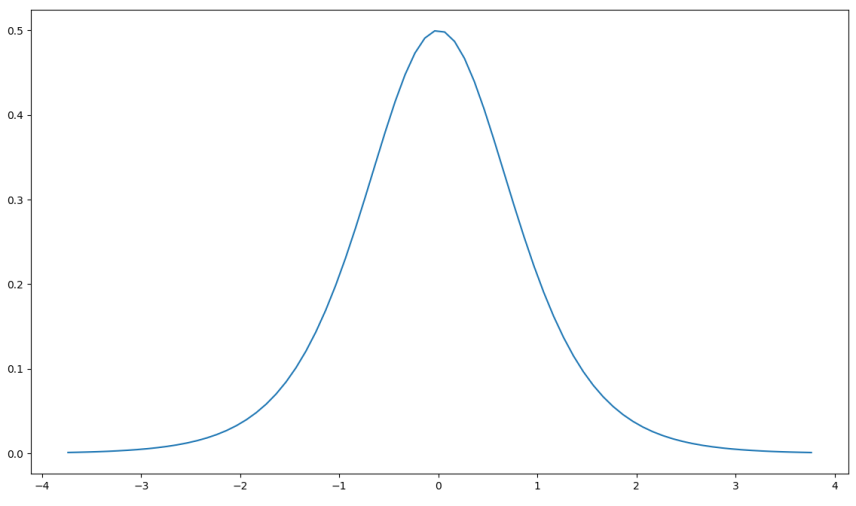


График функции плотности распределения 

1. **Код программы**

**import** math **as** mh  
**import** pandas **as** pd  
**import** numpy **as** np  
**import** matplotlib.pyplot **as** plt  
**from** pylab **import** \*  
**import** scipy  
**import** scipy.stats  
**from** scipy **import** integrate  
**from** scipy.stats **import** gamma  
**import** time  
**from** datetime **import** datetime  
**import** timeit  
**import** random  
  
pd.set\_option(**'display.max\_columns'**, **None**)  
  
  
**def** read\_data(): *# считывание данных из файла* data = []  
 **with** open(**"data.txt"**) **as** f:  
 **for** line **in** f:  
 data.append([float(x) **for** x **in** line.split()])  
 **return** data  
  
  
**def** seq\_gen(a, b, c, x\_t, N, m): *# генератор псевдослучайной последовательности* x = []  
 x\_n = 0  
 x\_p = 0  
  
 **for** i **in** range(N):  
 x\_n = (a \* x\_t + b \* x\_p + c) % m  
 x\_p = x\_t  
 x\_t = x\_n  
 x.insert(i, x\_n)  
 **return** x  
  
  
**def** logist(v, miu, x): *# функция плотности логистического распределения* **return** exp(-(x - miu) / v) / (v \* (1 + exp(- (x - miu) / v)) \*\* 2)  
  
  
**def** dens\_logist(v, miu, x): *# функция логистического распределения* **return** 1 / (1 + mh.exp(-(x - miu) / v))  
  
  
**def** cont\_seq(v, miu, n, a):*#генератор непрерывных псевдослучайных чисел* seq = []  
 **for** i **in** range(n):  
 seq.append(miu - v \* mh.log(1 / a[i] - 1, mh.e))  
 **return** seq  
  
  
**def** P\_crit(Sk, r): *# Вычисление значения статистики* coef = 1 / (np.power(2, (r / 2)) \* mh.gamma(r / 2))  
 func = **lambda** S: np.power(S, (r / 2 - 1)) \* (np.power(mh.e, (- S / 2)))  
 P = integrate.quad(func, Sk, np.inf)  
 print(P[0], coef)  
 **return** P[0] \* coef  
  
  
**def** hi\_2(x, alpha, V, miu):  
 print(**'Проверка критерия χ2:\n'**)  
 Max = np.max(x) *# максимально возможный элемент в последовательности длиной N* Min = np.min(x) *# минимально возможный элемент в последовательности длиной N* h = Max - Min *# длина отрезка интервалов* sum = 0  
 v = []  
 interval = [] *# список граничных точек интервала  
  
 # определяем число интервалов K* K = int(5 \* mh.log10(len(x)))  
  
 *# заполняем список нулями* **for** i **in** range(K + 1):  
 interval.append(0)  
  
 *# заполняем список граничными точками* u = Min  
 **for** i **in** range(K + 1):  
 interval[i] = u  
 u = interval[i] + h / K  
 P = []  
  
 print(**"Inter"**, interval)  
 *# заполняем список частот нулями* **for** i **in** range(K):  
 v.append(0)  
  
 *# подсчет частоты попаданий элементов выборки в интервалы* **for** i **in** range(len(x)):  
 **for** j **in** range(K):  
 **if** x[i] >= interval[j] **and** x[i] <= interval[j + 1]:  
 v[j] += 1  
  
 print(**"V = "**, v)  
 **for** i **in** range(len(v)):  
 v[i] /= len(x) *# отностельные частоты элементов* print(**"V = "**, v)  
 **for** i **in** range(1, len(interval)):  
 buf = dens\_logist(V, miu, interval[i]) - dens\_logist(V, miu, interval[i - 1])  
 P.append(buf)  
 print(**"P"**, P)  
  
 fig = plt.figure(figsize=(14, 8))  
 ax = fig.add\_subplot()  
 y1 = []  
 x1 = np.arange(Min, Max, 0.1)  
 **for** i **in** x1:  
 y1.append(dens\_logist(V, miu, i))  
 ax.plot(x1, y1) *# График* plt.show()  
  
 fig = plt.figure(figsize=(14, 8))  
 ax0 = fig.add\_subplot()  
 y2 = []  
 x2 = np.arange(Min, Max, 0.1)  
 **for** i **in** x2:  
 y2.append(logist(V, miu, i))  
 ax0.plot(x2, y2) *# График* plt.show()  
  
 *# график, построенный по группированным для критерия χ2 данным* gridsize = (1, 1)  
 fig = plt.figure(figsize=(14, 8))  
 ax1 = plt.subplot2grid(gridsize, (0, 0), colspan=2, rowspan=3)  
 plt.xlabel(**'Интервалы'**)  
 plt.ylabel(**'Частоты'**)  
 ax1.hist(interval[:-1], bins=interval, weights=v, rwidth=0.95)  
 ax1.set\_title(**'График, построеный по группированным для критерия χ2 данным'**)  
 plt.show()  
 *# вычисляем сумму в формуле S(hi2)* **for** i **in** range(K):  
 sum += mh.pow((v[i] - P[i]), 2) / P[i]  
 S = len(x) \* sum *# умножаем сумму на объем выборки n* r = K - 1 *# число степеней свободы* Sk = scipy.stats.chi2.ppf(1 - alpha, r)  
 print(S, r)  
 S\_star = P\_crit(S, r)  
 **if** S\_star > alpha:  
 print(**'P {Sχ2 > Sχ2\*}(достигнутый уровень значимости) = '**, S\_star)  
 print(**'Проверка критерия χ2 окончена.\n'**)  
 **return** 1, S, S\_star *# гипотеза отвергается* **else**:  
 print(**'P {Sχ2 > Sχ2\*}(достигнутый уровень значимости) = '**, S\_star)  
 print(**'Проверка критерия χ2 окончена.\n'**)  
 **return** 0, S, S\_star *# гипотеза не отвергается***def** I(v, z):  
 Iv = 0  
 **for** i **in** range(20):  
 I1 = (z / 2) \*\* (v + 2 \* i)  
 I1 /= mh.gamma(i + 1) \* mh.gamma(i + v + 1)  
 Iv += I1  
 **return** Iv  
  
  
**def** a1(s):  
 al = 0  
 **for** i **in** range(20):  
 arg = ((4 \* i + 1) \*\* 2) / (16 \* s)  
 a1 = math.gamma(i + 1 / 2) \* math.sqrt(4 \* i + 1)  
 a1 \*= math.exp(-arg)  
 a1 \*= I(-1 / 4, arg) - I(1 / 4, arg)  
 a1 /= math.gamma(1 / 2) \* math.gamma(i + 1)  
 al += a1  
 al \*= 1 / math.sqrt(2 \* s)  
 **return** al  
  
  
**def** O\_mega\_KMS(x, alpha, v, miu):  
 print(**'Проверка критерия Омега квадрат:\n'**)  
 x\_s = sorted(x) *# упорядочиваем выборку по возростанию* n = 0  
 S = 0 *# значение статистики критерия* P = 0 *# достигнутый уровень значимости* **for** i **in** range(1, len(x\_s)):  
 S += mh.pow(dens\_logist(v, miu, x\_s[i - 1]) - (2 \* i - 1) / (2 \* len(x)), 2)  
 S += 1 / (12 \* len(x))  
  
 P = 1 - a1(S)  
  
 **if** P > alpha:  
 print(**'P {S > S\*}(достигнутый уровень значимости) = '**, P)  
 print(**'Проверка критерия Омега квадрат окончена.\n'**)  
 **return** 1, S, P *# гипотеза не отвергается* **else**:  
 print(**'P {S > S\*}(достигнутый уровень значимости) = '**, P)  
 print(**'Проверка критерия Омега квадрат окончена.\n'**)  
 **return** 0, S, P *# гипотеза отвергается*data = read\_data()  
  
a = int(data[0][0])  
b = int(data[1][0])  
c = int(data[2][0])  
m = data[3][0]  
x = int(data[4][0])  
n1 = int(data[5][0])  
n2 = int(data[6][0])  
n3 = int(data[7][0])  
miu1 = data[8][0]  
miu2 = data[9][0]  
miu3 = data[10][0]  
v1 = data[11][0]  
v2 = data[12][0]  
v3 = data[13][0]  
alpha = data[14][0]  
  
N = []  
V = []  
MIU = []  
test = [1.2, 3, 12, 22, 23, 10, 14]  
**for** i **in** range(5, 8):  
 N.append(int(data[i][0]))  
  
**for** i **in** range(8, 11):  
 MIU.append(data[i][0])  
  
**for** i **in** range(11, 14):  
 V.append(data[i][0])  
  
**for** i **in** range(len(N)):  
 **for** j **in** range(len(V)):  
 alp = seq\_gen(a, b, c, x, N[i], m)  
 start = timeit.default\_timer()  
 s1 = cont\_seq(V[j], MIU[j], N[i], alp)  
  
 end = timeit.default\_timer() - start  
 print(**"Для моделирования выборки с параметрами: u ="**, MIU[j], **" v ="**, V[j], **" потребовалось "**, end, **" секунд"**)  
 h, s, S = hi\_2(s1, alpha, V[j], MIU[j])  
 O, O\_s, P = O\_mega\_KMS(s1, alpha, V[j], MIU[j])  
 **if** i == 0:  
 **if** j == 0:  
 f = open(**"out\_seq\_50\_1.txt"**, **"w"**)  
 **if** j == 1:  
 f = open(**"out\_seq\_50\_2.txt"**, **"w"**)  
 **if** j == 2:  
 f = open(**"out\_seq\_50\_3.txt"**, **"w"**)  
 **if** i == 1:  
 **if** j == 0:  
 f = open(**"out\_seq\_200\_1.txt"**, **"w"**)  
 **if** j == 1:  
 f = open(**"out\_seq\_200\_2.txt"**, **"w"**)  
 **if** j == 2:  
 f = open(**"out\_seq\_200\_3.txt"**, **"w"**)  
 **if** i == 2:  
 **if** j == 0:  
 f = open(**"out\_seq\_1000\_1.txt"**, **"w"**)  
 **if** j == 1:  
 f = open(**"out\_seq\_1000\_2.txt"**, **"w"**)  
 **if** j == 2:  
 f = open(**"out\_seq\_1000\_3.txt"**, **"w"**)  
 p = **'Последовательность: '** f.write(p)  
 **for** u **in** s1:  
 p = str(round(u, 3)) + **', '** f.write(p)  
 p = **'\nВремея моделирования: '** + str(end) + **' секунд\n'** f.write(p)  
 **"""p = 'Параметры последовательности: \n'  
 p = 'n = ' + str(N[i]) + 'miu = ' + str(MIU[j]) + 'v = ' + str(V[j]) + '\n'  
 f.write(p)"""** p = **'Значение статистики S: '** + str(O\_s) + **'\n'** f.write(p)  
 p = **'Достигнутый уровень значимости: '** + str(P) + **'\n'** f.write(p)  
 **if** O:  
 p = **'Критери омега^2-Крамера-Мизеса-Смирнова пройден, следовательно не причин для отклонения гипотезы о согласии распределения сгенерированной последовательности с логистическим распределением'** + **'\n'** f.write(p)  
 **else**:  
 p = **'Критери омега^2-Крамера-Мизеса-Смирнова, следовательно гипотеза о согласии распределения сгенерированной последовательности с логистическим распределением отвергается'** + **'\n'** f.write(p)  
  
 p = **'Значение статистики S: '** + str(s) + **'\n'** f.write(p)  
 p = **'Достигнутый уровень значимости: '** + str(S) + **'\n'** f.write(p)  
 **if** h:  
 p = **'Критери Xi^2 пройден, следовательно не причин для отклонения гипотезы о согласии распределения сгенерированной последовательности с логистическим распределением'** + **'\n'** f.write(p)  
 **else**:  
 p = **'Критери Xi^2 не пройден, следовательно гипотеза о согласии распределения сгенерированной последовательности с логистическим распределением отвергается'** + **'\n'** f.write(p)  
 f.close()

1. **Выводы**

В ходе проведённой работы, было разработано программное средство для генерации последовательностей псевдослучайных непрерывных величин. Было проведено моделирование последовательностей псевдослучайных величин на разных наборах входных данных по формуле. Также были проверены критерии χ2 и ω2-Крамера-Мизеса-Смирнова.