

Домашнее задание №2 по курсу математической статистики.

Шубин Н. В. СКБ 172

Моделирование логарифмического распределения.

Написав программу для моделирования логарифмического распределения, были смоделированы 25 различных выборок, по 5 выборок одной длины для 5 различных длин: 5, 10, 100, 1000, 10000.

Для длины $n = 5$:

[2, 2, 1, 1, 2]

[2, 1, 2, 1, 1]

[1, 1, 1, 1, 1]

[1, 1, 1, 1, 1]

[4, 2, 1, 6, 1]

Для длины $n=10$:

[1, 2, 11, 1, 1, 1, 2, 1, 2, 1]

[4, 1, 2, 1, 1, 4, 1, 2, 2, 2]

[2, 1, 1, 1, 1, 2, 1, 1, 1, 1]

[2, 3, 2, 1, 3, 1, 5, 3, 1, 2]

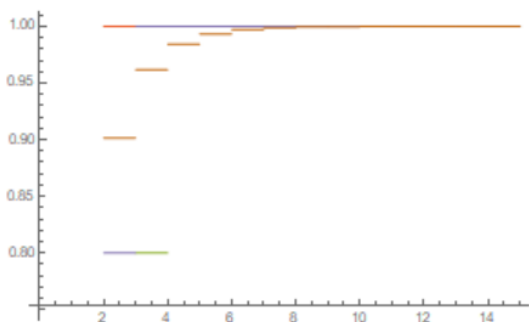
[3, 5, 1, 1, 3, 1, 1, 8, 7, 1]

Графики эмпирической функции распределения и функции распределения.

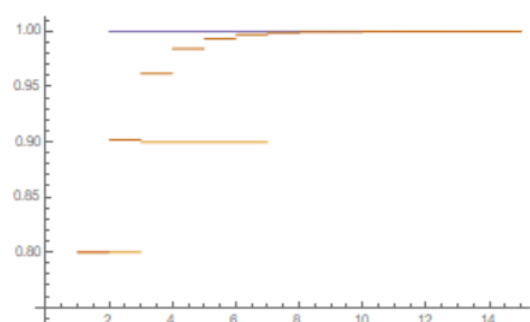
Построения выполнены в Wolfram Mathematica, основываясь на сгенерированных выборках.

Сравнены с теоретической функцией распределения.

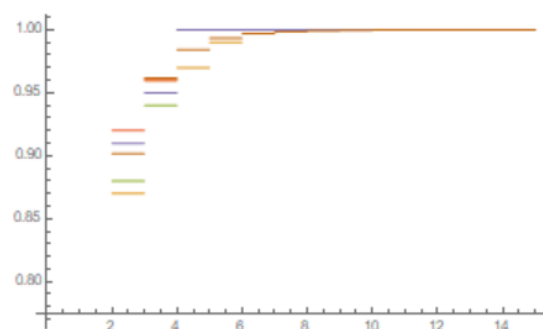
Для $n = 5$



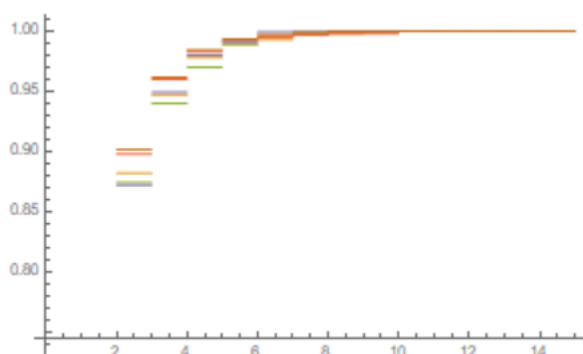
Для $n = 10$



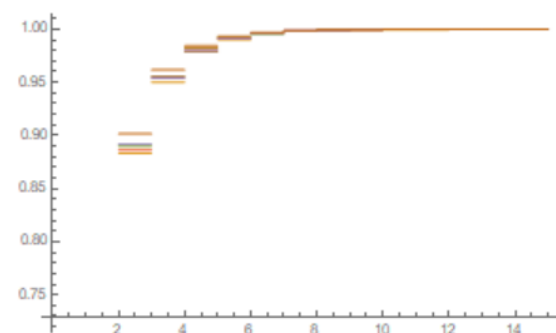
Для $n = 100$



Для $n = 1000$



Для $n = 10000$



Моделирование распределения Эрланга.

Также как и для логарифмического распределения была написана программа для выборок размера: 5, 10, 100, 1000, 10000.

Для $n = 5$

Er5_1 – Блокнот	–	□
Файл Правка Формат Вид Справка		
22.91860914551629 37.36648484065653 25.13281377646234 25.85479819486912 34.006483122726365		
Er5_2 – Блокнот	–	
Файл Правка Формат Вид Справка		
28.70262336947717 30.14450702900026 31.963642591254025 24.22686300756681 20.6597455553605		
Er5_3 – Блокнот		
Файл Правка Формат Вид Справка		
32.405613939933644 29.890395994637863 40.45385234861331 30.718033294522584 36.59484237216847		
Er5_4 – Блокнот		
Файл Правка Формат Вид Справка		
29.229757966906632 33.492058772416776 38.13731601382244 28.758024349282287 24.950633898051905		
Er5_5 – Блокнот		
Файл Правка Формат Вид Справка		
35.113868071976455 25.58680432450295 33.791424517319896 27.677671163045126 39.9050378719883		

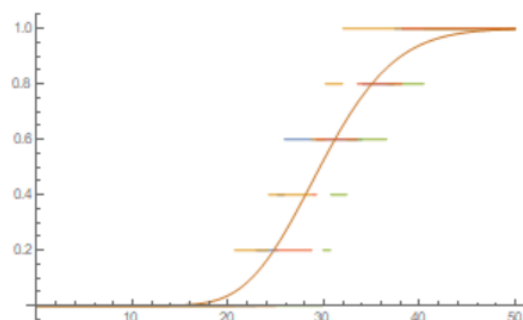
Для $n = 10$:

*Er10_1 – Блокнот	
Файл Правка Формат Вид Справка	
30.686500627212716 36.86042659365773 29.015818762837576 30.29533067355636 25.92264389648561 33.46965516557787 22.730982789171748 33.821533777509785 42.22411350959774 22.235256640242593	
*Er10_2 – Блокнот	
Файл Правка Формат Вид Справка	
26.79175306761235 29.604222806793373 32.90973956380678 31.795314048544213 22.087672731911432 33.29812241031793 40.953571655977846 34.304836571542005 43.36247010083019 34.102314016627965	
*Er10_3 – Блокнот	
Файл Правка Формат Вид Справка	
26.001720840855636 30.55725042842186 37.735253468157865 33.240702951812516 32.97851793121551 25.006955863064412 26.540012603653405 31.059799932791858 29.88970905894295 31.676494325302613	
*Er10_4 – Блокнот	
Файл Правка Формат Вид Справка	
31.569495316455573 27.89503664807224 28.097922326344378 23.957416441774512 27.43626899446591 31.492359559275947 31.551214421255178 33.10137873139289 22.89676037428023 26.842877563806518	
*Er10_5 – Блокнот	
Файл Правка Формат Вид Справка	
36.83980201290634 28.52314537437814 28.043463328480602 29.89781245889276 26.123574823153074 22.994319463805553 38.84899667703125 32.39100679706483 30.12740206885096 41.99675284056126	

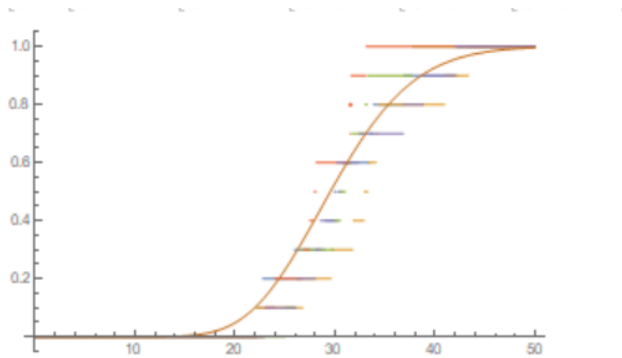
Графики так же построены в Wolfram Mathematica.

Сравнены с теоретической функцией распределения

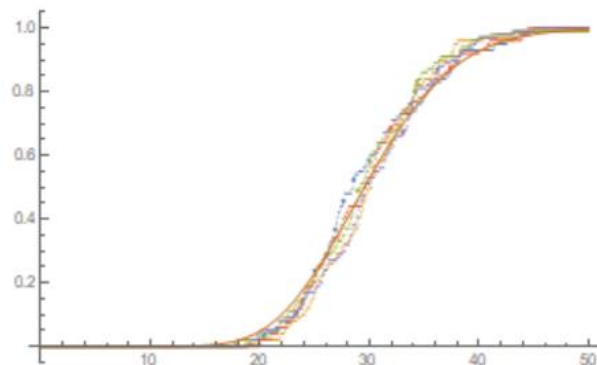
Для $n = 5$



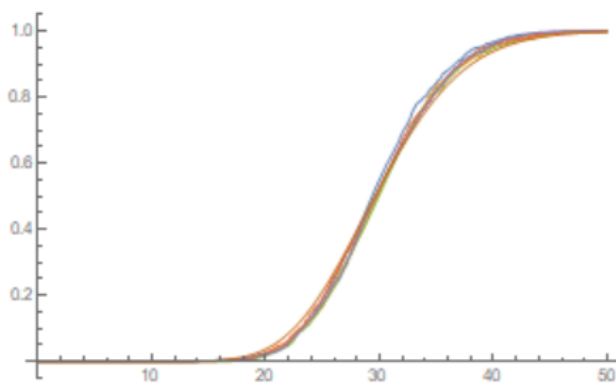
Для $n = 10$



Для $n = 100$



Для $n = 1000$



3. Построение вариационного ряда выборки.

3.1. Логарифмическое распределение.

Построение вариационного ряда и расчет квантилей.

```

var = []

file = open('log5_1.txt', 'r')
data = file.read()
data = ''.join(data)
data = data.split()
var.append(sorted(list(map(float, data))))
print(data, '\n')

file = open('log5_2.txt', 'r')
data = file.read()
data = ''.join(data)
data = data.split()
var.append(sorted(list(map(float, data))))
print(data, '\n')

file = open('log5_3.txt', 'r')
data = file.read()
data = ''.join(data)
data = data.split()
var.append(sorted(list(map(float, data))))
print(data, '\n')

file = open('log5_4.txt', 'r')
data = file.read()
data = ''.join(data)
data = data.split()
var.append(sorted(list(map(float, data))))
print(data, '\n')

file = open('log5_5.txt', 'r')
data = file.read()
data = ''.join(data)
data = data.split()
var.append(sorted(list(map(float, data))))
print(data, '\n')

```

```

def quantils(data, f, j, k):
    quant1 = []
    quant2 = []
    quant3 = []
    n1 = int(f * len(data[1]) + 1)
    n2 = int(j * len(data[1]) + 1)
    n3 = int(k * len(data[1]) + 1)
    for i in range(5):
        quant1.append(data[i][n1-1])
    for i in range(5):
        quant2.append(data[i][n2-1])
    for i in range(5):
        quant3.append(data[i][n3-1])
    print('Квантили уровня {} для n = {}'.format(f, len(data[1])), quant1)
    print('Квантили уровня {} для n = {}'.format(j, len(data[1])), quant2)
    print('Квантили уровня {} для n = {}'.format(k, len(data[1])), quant3)

```

Для n = 5:

['1', '1', '1', '1', '1']

['2', '1', '2', '2', '2']

['2', '4', '2', '2', '3']

['1', '2', '2', '2', '1']

['1', '3', '2', '1', '1']

Квантили уровня 0.1 для n = 5 [1.0, 1.0, 2.0, 1.0, 1.0]

Квантили уровня 0.5 для n = 5 [1.0, 2.0, 2.0, 2.0, 1.0]

Квантили уровня 0.7 для n = 5 [1.0, 2.0, 3.0, 2.0, 2.0]

Для n = 10:

['2', '1', '1', '2', '1', '1', '1', '1', '1', '1']

['7', '1', '1', '2', '1', '3', '1', '1', '2', '2']

['1', '1', '1', '1', '1', '1', '2', '1', '2', '1']

['1', '1', '1', '2', '1', '1', '2', '1', '1', '1']

['1', '1', '1', '2', '1', '2', '2', '1', '1', '1']

Квантили уровня 0.1 для n = 10 [1.0, 1.0, 1.0, 1.0, 1.0]

Квантили уровня 0.5 для n = 10 [1.0, 2.0, 1.0, 1.0, 1.0]

Квантили уровня 0.7 для n = 10 [1.0, 2.0, 1.0, 1.0, 2.0]

Для $n = 100$:

Квантили уровня 0.1 для $n = 100$ [1.0, 1.0, 1.0, 1.0, 1.0]

Квантили уровня 0.5 для $n = 100$ [1.0, 1.0, 1.0, 1.0, 1.0]

Квантили уровня 0.7 для $n = 100$ [2.0, 2.0, 2.0, 1.0, 2.0]

Для $n = 1000$:

Квантили уровня 0.1 для $n = 1000$ [1.0, 1.0, 1.0, 1.0, 1.0]

Квантили уровня 0.5 для $n = 1000$ [1.0, 1.0, 1.0, 1.0, 1.0]

Квантили уровня 0.7 для $n = 1000$ [2.0, 2.0, 2.0, 1.0, 2.0]

Нахождение разностей эмпирических функций.

```
import numpy as np
import math
smp = [5, 10, 100, 1000, 10000]
data = []
def same(k):
    mas = []
    for i in k:
        if i not in mas:
            mas.append(i)
    return(mas)
def f(a,b):
    s = 0
    for i in range(len(a)):
        if a[i] < b:
            s += 1
        else:
            break
    return(s / len(a))
for k1 in range(len(smp)):
    var = []
    maxs = []
    for i in range(len(smp)):
        file = open('log{}.txt'.format(smp[k1], i+1), 'r')
        data = file.read()
        data = ''.join(data)
        data = data.split()
        var.append(sorted(list(map(float, data))))
    for i in range(4):
        for j in range(i+1, 5):
            max = -100000
            mas = sorted(np.concatenate([var[i], var[j]]))
            for k in range(len(mas)):
                if math.fabs(f(var[i], mas[k]) - f(var[j], mas[k])) > max:
                    max = math.fabs(f(var[i], mas[k]) - f(var[j], mas[k]))
            maxs.append(max)
    print("\n = {}".format(smp[k1]), maxs, '\n')
```

$n = 5$ [0.8, 1.0, 0.6, 0.4, 0.4, 0.2, 0.39999999999999997, 0.4, 0.6, 0.19999999999999996]

$n = 10$ [0.30000000000000004, 0.0, 0.0, 0.10000000000000009, 0.30000000000000004,
0.30000000000000004, 0.19999999999999996, 0.0, 0.10000000000000009, 0.10000000000000009]

$n = 100$ [0.020000000000000018, 0.020000000000000018, 0.06000000000000005,
0.050000000000000044, 0.030000000000000027, 0.06999999999999995, 0.040000000000000036,
0.050000000000000044, 0.039999999999999925, 0.08999999999999997]

$n = 1000$ [0.013000000000000012, 0.02100000000000002, 0.015000000000000013,
0.026999999999999913, 0.013999999999999901, 0.016000000000000014, 0.03299999999999992,
0.024000000000000002, 0.019000000000000017, 0.041999999999999926]

3.2 Распределение Эрланга.

Построение вариационного ряда выборки и расчет квантилей по предыдущему коду.

Для $n = 5$:

['22.91860914551629', '37.36648484065653', '25.13281377646234', '25.85479819486912', '34.006483122726365']
['28.70262336947717', '30.14450702900026', '31.963642591254025', '24.22686300756681', '20.65974555553605']
['32.405613939933644', '29.890395994637863', '40.45385234861331', '30.718033294522584', '36.59484237216847']
['29.229757966906632', '33.492058772416776', '38.13731601382244', '28.758024349282287', '24.950633898051905']
['35.113868071976455', '25.58680432450295', '33.791424517319896', '27.677671163045126', '39.9050378719883']

Квантили уровня 0.1 для $n = 5$ [22.91860914551629, 20.65974555553605, 29.890395994637863, 24.950633898051905, 25.58680432450295]

Квантили уровня 0.5 для $n = 5$ [25.85479819486912, 28.70262336947717, 32.405613939933644, 29.229757966906632, 33.791424517319896]

Квантили уровня 0.7 для $n = 5$ [34.006483122726365, 30.14450702900026, 36.59484237216847, 33.492058772416776, 35.113868071976455]

Для $n = 10$:

['30.686500627212716', '36.86042659365773', '29.015818762837576', '30.29533067355636', '25.92264389648561', '33.46965516557787', '22.730982789171748', '33.821533777509785', '42.22411350959774', '22.235256640242593']
['26.79175306761235', '29.604222806793373', '32.90973956380678', '31.795314048544213', '22.087672731911432', '33.29812241031793', '40.953571655977846', '34.304836571542005', '43.36247010083019', '34.102314016627965']
['26.001720840855636', '30.55725042842186', '37.735253468157865', '33.240702951812516', '32.97851793121551', '25.006955863064412', '26.540012603653405', '31.059799932791858', '29.88970905894295', '31.676494325302613']
['31.569495316455573', '27.89503664807224', '28.097922326344378', '23.957416441774512', '27.43626899446591', '31.492359559275947', '31.551214421255178', '33.10137873139289', '22.89676037428023', '26.842877563806518']
['36.83980201290634', '28.52314537437814', '28.043463328480602', '29.89781245889276', '26.123574823153074', '22.994319463805553', '38.84899667703125', '32.39100679706483', '30.12740206885096', '41.99675284056126']

Квантили уровня 0.1 для $n = 10$ [22.730982789171748, 26.79175306761235, 26.001720840855636, 23.957416441774512, 26.123574823153074]

Квантили уровня 0.5 для $n = 10$ [30.686500627212716, 33.29812241031793, 31.059799932791858, 28.097922326344378, 30.12740206885096]

Квантили уровня 0.7 для $n = 10$ [33.821533777509785, 34.304836571542005, 32.97851793121551, 31.551214421255178, 36.83980201290634]

Для $n = 100$

Квантили уровня 0.1 для $n = 100$ [23.36193159382023, 24.122505071143916, 22.562129590506157, 23.041570470046235, 23.245448988508578]

Квантили уровня 0.5 для $n = 100$ [28.473916156161884, 30.228015514855027, 29.215061506867425, 29.650706942238113, 29.703218874825332]

Квантили уровня 0.7 для $n = 100$ [32.007624354219836, 32.86463625135326, 32.924871295389174, 32.65700874131317, 33.538266031184996]

Для $n = 1000$:

Квантили уровня 0.1 для $n = 1000$ [23.183649154845718, 23.23285362958481, 23.396619051168905, 23.044476571316014, 23.338024105387667]

Квантили уровня 0.5 для $n = 1000$ [29.386462625612396, 29.78411956842921, 29.996212753304857, 29.73241609326575, 29.635818958562968]

Квантили уровня 0.7 для $n = 1000$ [32.1265044923083, 32.75615543666772, 32.69468425761642, 32.474439343803354, 32.737360269039755]

Нахождение разностей эмпирических функций.

$n = 5$ [0.4, 0.6, 0.39999999999999997, 0.4, 0.60000000000000001, 0.4, 0.6, 0.6, 0.4, 0.4]

$n = 10$ [0.3, 0.30000000000000004, 0.4, 0.2, 0.4, 0.60000000000000001, 0.3, 0.30000000000000004, 0.20000000000000007, 0.30000000000000004]

$n = 100$ [0.18000000000000005, 0.13, 0.11000000000000004, 0.18, 0.12, 0.10999999999999999, 0.07, 0.07999999999999996, 0.08999999999999997, 0.12]

$n = 1000$ [0.058000000000000005, 0.054999999999999994, 0.0380000000000000034, 0.048000000000000004, 0.024000000000000002, 0.0310000000000000028, 0.022000000000000002, 0.042000000000000004, 0.032000000000000003, 0.03999999999999998]

4. Построение гистограмм и полигон частот.