



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«МИРЭА – Российский технологический университет»

ИНСТИТУТ КИБЕРНЕТИКИ
КАФЕДРА ВЫСШЕЙ МАТЕМАТИКИ

Лабораторная работа 1
по курсу «Теория вероятностей и математическая статистика, часть 2»

ВАРИАНТ 63

Тема: Первичная обработка выборки из
дискретной генеральной совокупности

Выполнила:
Студентка 3-го курса
Макарчук Н.С.

Группа: КМБО-03-18

1 Задание

Задание 1. Получить выборку, сгенерировав 200 псевдослучайных чисел, распределенных по биномиальному закону с параметрами n и p .

$$n = 7 + V \bmod 15; p = 0,2 + 0,005V$$

Задание 2. Получить выборку, сгенерировав 200 псевдослучайных чисел, распределенных по геометрическому закону с параметром. $p = 0,2 + 0,005V$

Задание 3. Получить выборку, сгенерировав 200 псевдослучайных чисел, распределенных по закону Пуассона с параметром. $\lambda = 1 + 0,02V$

Следуя **Указаниям** для всех выборок построить:

- 1) статистический ряд;
- 2) полигон относительных частот;
- 3) график эмпирической функции распределения;

Найти:

- 1) выборочное среднее;
- 2) выборочную дисперсию;
- 3) выборочное среднее квадратическое отклонение;
- 4) выборочную моду;
- 5) выборочную медиану;
- 6) выборочный коэффициент асимметрии;
- 7) выборочный коэффициент эксцесса.

Провести сравнение рассчитанных характеристик с теоретическими значениями. V - номер варианта. Вычисления проводить с точностью до 0,00001.

2 Краткие теоретические сведения

2.1 Биномиальное распределение

- Ряд распределения

x_i	0	1	2	...	m	...	n
p_i	q^n	$C_n^1 p q^{n-1}$	$C_n^2 p^2 q^{n-2}$...	$C_n^m p^m q^{n-m}$...	p^n

- Математическое ожидание

$$M(X) = np$$

- Дисперсия

$$D(X) = npq, \text{ где } q = 1 - p$$

- Среднее квадратичное отклонение

$$\sigma = \sqrt{D(X)} = \sqrt{npq}$$

- Мода

$$M_o = \begin{cases} [(n+1)p], \text{ если } (n+1)p - \text{дробное;} \\ (n+1)p - \frac{1}{2}, \text{ если } (n+1)p - \text{целое.} \end{cases}$$

- Медиана $M_e = \text{Round}(np)$

- Коэффициент асимметрии $a_s = \gamma_1 = \frac{q-p}{\sqrt{npq}}$

- Коэффициент эксцесса $\epsilon_k = \gamma_2 = \frac{1-6pq}{npq}$

2.1 Геометрическое распределение

- Ряд распределения

x_i	0	1	2	...	m	...
p_i	p	qp	q^2p	...	$q^m p$...

- Математическое ожидание

$$M(X) = \frac{q}{p}, \text{ где } q=1-p$$

- Дисперсия

$$D(X) = \frac{q}{p^2}, \text{ где } q = 1 - p$$

- Среднее квадратичное отклонение

$$\sigma = \sqrt{D(X)} = \frac{\sqrt{q}}{p}$$

- Мода $M_o = 0$

- Медиана

$$M_e = \begin{cases} \left[-\frac{\ln 2}{\ln q} \right], & \text{если } \frac{\ln 2}{\ln q} - \text{дробное} \\ -\frac{\ln 2}{\ln q} - \frac{1}{2}, & \text{если } \frac{\ln 2}{\ln q} - \text{целое} \end{cases}$$

- Коэффициент асимметрии

$$a_s = \gamma_1 = \frac{2-p}{\sqrt{q}}$$

- Коэффициент эксцесса

$$\varepsilon_k = \gamma_2 = 6 + \frac{p^2}{q}$$

2.3 Распределение Пуассона

- Ряд распределения

x_i	0	1	2	...	m	...
p_i	$e^{-\lambda}$	$\lambda e^{-\lambda}$	$\frac{\lambda^2 e^{-\lambda}}{2!}$...	$\frac{\lambda^m e^{-\lambda}}{m!}$...

- Математическое ожидание

$$M(X) = \lambda$$

- Дисперсия

$$D(X) = \lambda$$

- Среднее квадратичное отклонение

$$\sigma = \sqrt{D(X)} = \sqrt{\lambda}$$

- Мода

$$M_o = [\lambda]$$

- Медиана

$$M_e \approx \left[\lambda + \frac{1}{3} - \frac{0.02}{\lambda} \right]$$

- Коэффициент асимметрии

$$a_s = \gamma_1 = \lambda^{-\frac{1}{2}}$$

- Коэффициент эксцесса

$$\varepsilon_k = \gamma_2 = \frac{1}{\lambda}$$

2.4 Средства языка программирования

В программе расчёта был использован язык программирования Python. Работа осуществлялась в среде Jupyter Notebook.

Модуль Numpy:

Функции, которые использовались для генерации выборки

- 1) `random.geometric(p, N)` - генерирует N чисел по геометрическому распределению с параметром p — вероятность
- 2) `random.poisson(lambda, N)` — генерирует N чисел по распределению Пуассона, с параметром `lamda`.
- 3) `random.binomial(n, p, N)` — функция генерирует N чисел по Биноминальному распределению с параметрами n и p

`matplotlib.pyplot.plot(xs, ys)` – функция для отрисовки полигона частот:

3 Результаты расчётов

3.1 Задание 1

Биномиальная распределение

Параметры: $p = 0.515$, $n = 10$

▪ Полученная выборка

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

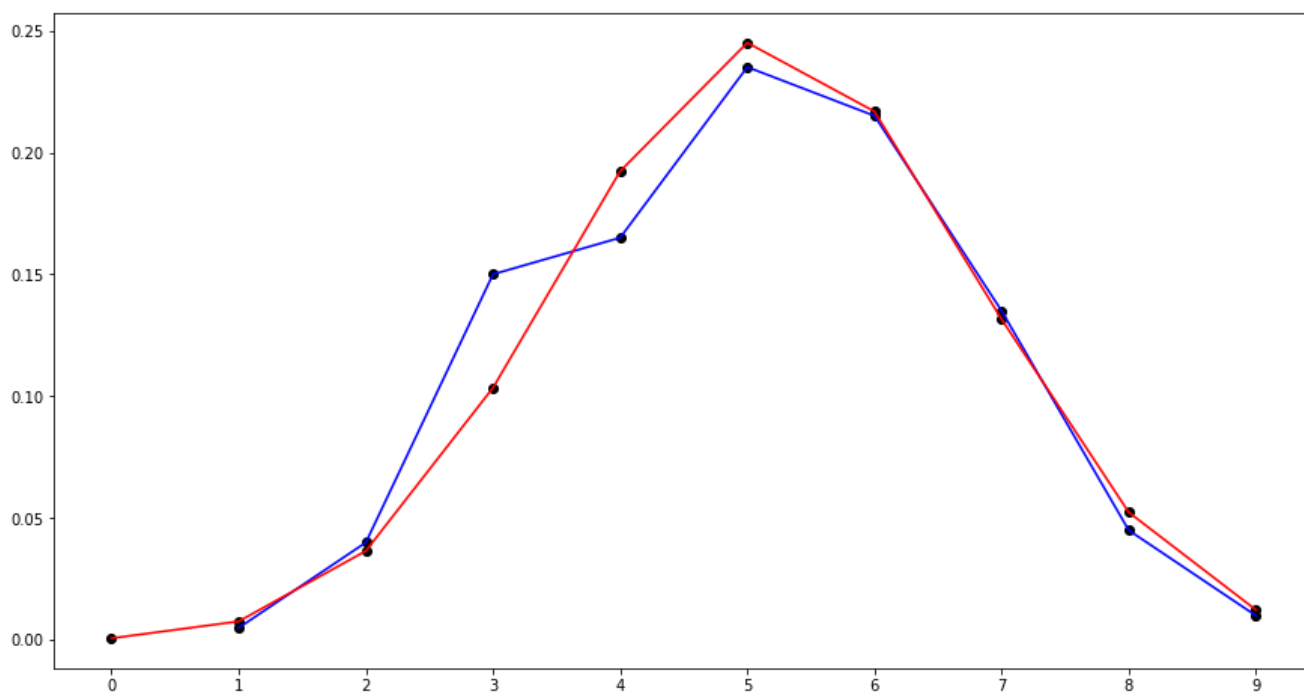
Упорядоченная выборка

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

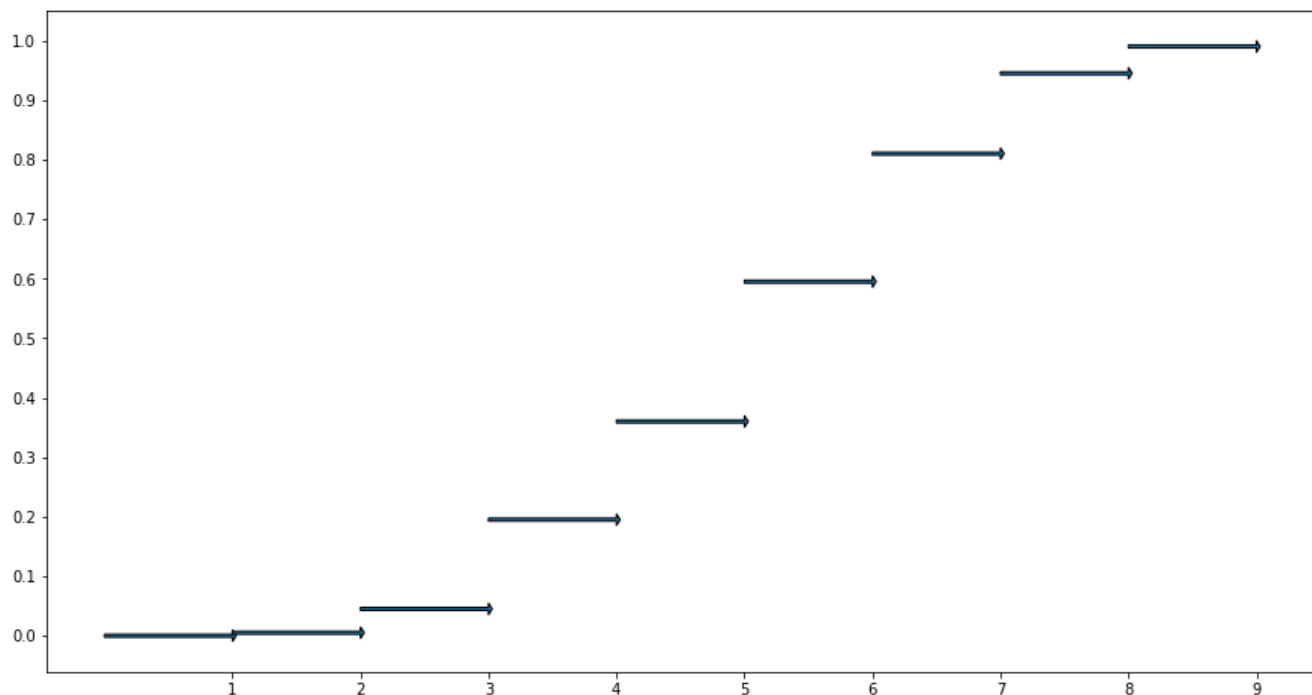
- Статистический ряд

x_i^*	n_i	W_i	S_i
1	1	0.005	0.005
2	8	0.04	0.045
3	30	0.15	0.195
4	33	0.165	0.36
5	47	0.235	0.595
6	43	0.215	0.81
7	27	0.135	0.945
8	9	0.045	0.99
9	2	0.01	1
	200		

- График полигона относительных частот



- Эмпирическая функция распределения



- Результаты расчетов требуемых характеристик

Выборочное среднее: 5.055

Выборочная дисперсия: 2.52198

Выборочное среднее квадратичное отклонение: 1.58807

Выборочная мода: 5

Выборочная медиана: 5

Выборочный коэффициент асимметрии: -0.00781

Выборочный коэффициент асимметрии: -0.54320

3.2 Задание 2

Геометрическое распределение

Полученная выборка

0	2	0	0	1	0	0	0	0
0	0	1	1	0	1	1	2	0
1	1	0	1	1	1	2	0	1
0	0	0	2	0	3	2	1	0
1	0	0	0	0	0	0	0	3
2	0	0	0	1	0	1	0	1
2	1	1	0	1	2	0	1	1
1	0	0	1	1	4	0	1	0
2	0	0	4	0	3	1	0	3
0	0	2	0	1	0	1	1	0
1	1	2	7	1	3	0	0	1
3	0	0	1	0	0	1	0	3
0	1	0	0	0	0	2	0	0
2	1	2	0	0	0	1	0	0
1	0	5	1	1	0	1	0	2
0	1	0	1	0	2	0	1	2
0	3	0	1	0	1	0	0	2
3	3	4	2	2	0	1	1	0
0	0	2	0	2	0	0	0	0
0	0	0	1	3	1	0	1	2

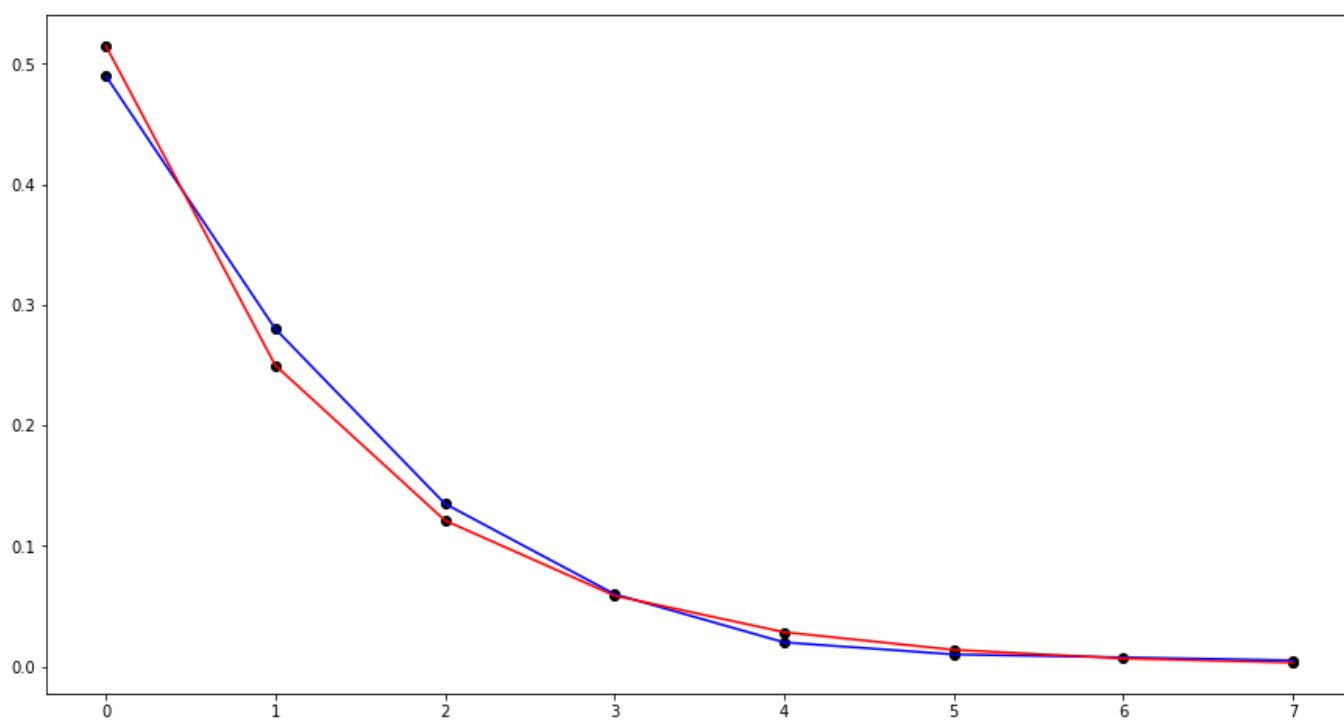
- Упорядоченная выборка

0	0	0	0	0	1	1	2	2
0	0	0	0	0	1	1	2	3
0	0	0	0	0	1	1	2	3
0	0	0	0	0	1	1	2	3
0	0	0	0	0	1	1	2	3
0	0	0	0	0	1	1	2	3
0	0	0	0	0	1	1	2	3
0	0	0	0	0	1	1	2	3
0	0	0	0	0	1	1	2	3
0	0	0	0	0	1	1	2	3
0	0	0	0	0	1	1	2	3
0	0	0	0	0	1	1	2	3
0	0	0	0	0	1	1	2	3
0	0	0	0	0	1	1	2	3
0	0	0	0	0	1	1	2	3
0	0	0	0	0	1	1	2	3
0	0	0	0	0	1	1	2	3
0	0	0	0	0	1	1	2	3
0	0	0	0	0	1	1	2	3
0	0	0	0	0	1	1	2	4
0	0	0	0	0	1	1	2	4
0	0	0	0	0	1	1	2	4
0	0	0	0	0	1	1	2	4
0	0	0	0	0	1	1	2	5
0	0	0	0	1	1	1	2	5
0	0	0	0	1	1	1	2	7

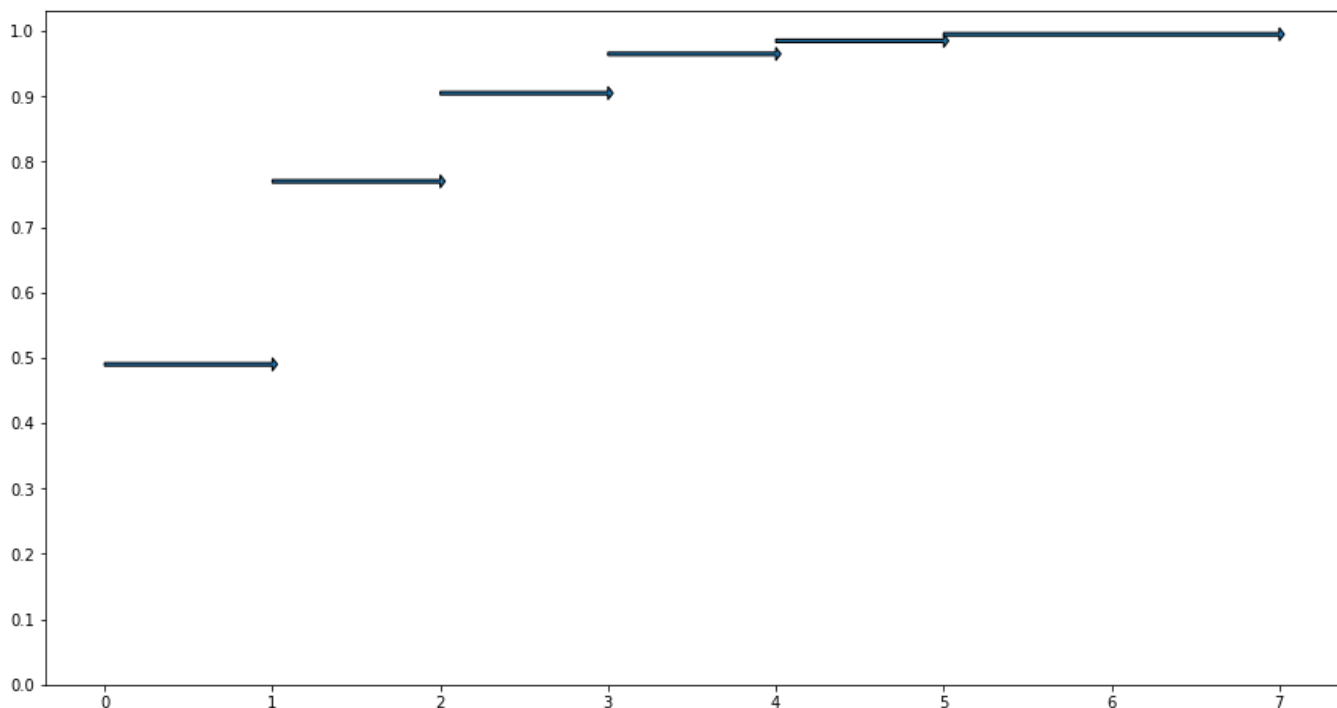
■ Статистический ряд

x_i^*	n_i	w_i	S_i
0	98	0.49	0.49
1	56	0.28	0.77
2	27	0.135	0.905
3	12	0.06	0.965
4	4	0.02	0.985
5	2	0.01	0.995
7	1	0.005	1
	200	1.0	-

■ График полигона относительных частот



- Эмпирическая функция распределения



Результаты расчетов требуемых характеристик

Выборочное среднее: 0.895

Выборочная дисперсия: 1.37397

Выборочное среднее квадратичное отклонение: 1.17216

Выборочная мода: 0

Выборочная медиана: 1

Выборочный коэффициент асимметрии: 1.75033

Выборочный коэффициент асимметрии: 4.065314

3.3 Задание 3

- Полученная выборка

2	4	3	3	1	2	3	1	1
3	3	1	4	2	2	0	3	2
1	5	3	0	1	0	0	0	1
5	0	3	3	2	2	5	2	3
3	3	0	2	4	3	2	2	3
3	2	1	4	2	4	0	2	1
3	2	2	0	3	1	2	1	2
2	1	0	0	2	3	4	2	2
1	2	5	3	4	4	1	2	1
3	2	2	2	1	2	1	1	3
2	3	2	6	2	0	1	4	4
4	1	6	3	2	2	2	0	2
3	1	2	5	2	1	6	7	1
1	3	0	2	6	5	3	6	2
1	1	2	0	3	4	1	2	3
1	2	3	1	2	2	3	0	3
0	2	4	2	1	4	3	2	3
2	1	3	0	4	1	1	2	1
2	0	1	4	3	0	2	3	2
6	2	1	2	1	2	4	3	3

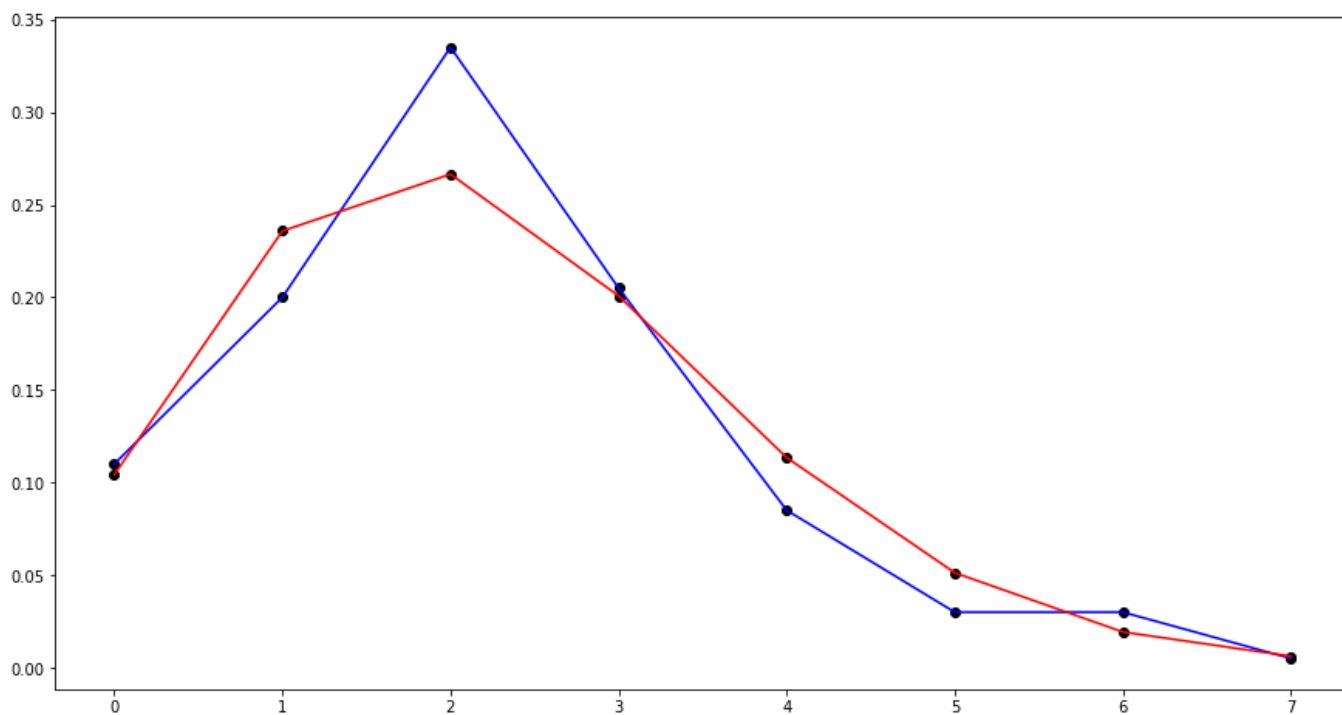
▪ Упорядоченная выборка

0	0	1	1	2	2	2	3	4
0	0	1	1	2	2	2	3	4
0	1	1	2	2	2	2	3	4
0	1	1	2	2	2	2	3	4
0	1	1	2	2	2	2	3	4
0	1	1	2	2	2	2	3	4
0	1	1	2	2	2	2	3	4
0	1	1	2	2	2	2	3	5
0	1	1	2	2	2	2	3	5
0	1	1	2	2	2	3	3	5
0	1	1	2	2	2	3	4	5
0	1	1	2	2	2	3	4	5
0	1	1	2	2	2	3	4	5
0	1	1	2	2	2	3	4	6
0	1	1	2	2	2	3	4	6
0	1	1	2	2	2	3	4	6
0	1	1	2	2	2	3	4	6
0	1	1	2	2	2	3	4	6
0	1	1	2	2	2	3	4	6
0	1	1	2	2	2	3	4	7

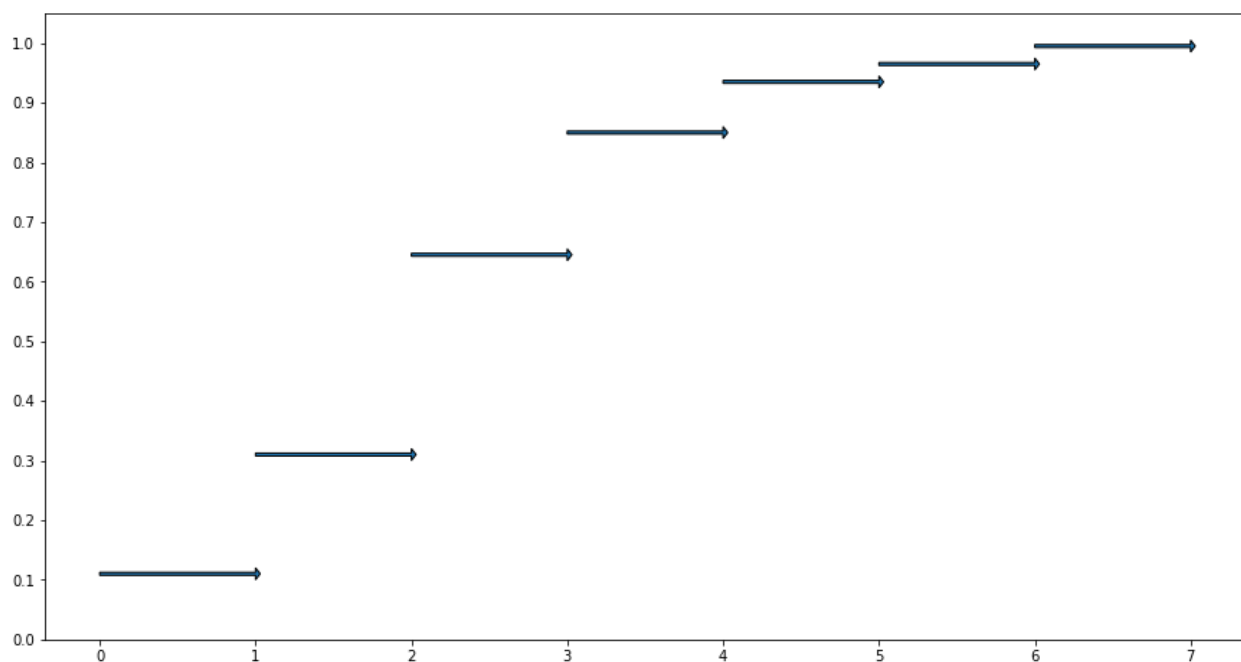
■ Статистический ряд

x_i^*	n_i	w_i	S_i
0	22	0.11	0.11
1	40	0.2	0.31
2	67	0.335	0.645
3	41	0.205	0.85
4	17	0.085	0.935
5	6	0.03	0.965
6	6	0.03	0.995
7	1	0.005	1
	200	1	-

■ График полигона относительных частот



- Эмпирическая функция распределения



Результаты расчетов требуемых характеристик

Выборочное среднее: 2.19

Выборочная дисперсия: 2.02389

Выборочное среднее квадратичное отклонение: 1.422638

Выборочная мода: 2

Выборочная медиана: 2

Выборочный коэффициент асимметрии: 0.694451

Выборочный коэффициент асимметрии: 0.613297

4 Анализ результатов и выводы

4.1 Таблица сравнения относительных частот и теоретических вероятностей

- Биномиальное распределение

j	\tilde{w}_j	p_j	$ \tilde{w}_j - p_j $
0	0	0.000720	0.00072
1	0.005	0.007647	0.002647
2	0.04	0.036539	0.003461
3	0.15	0.103465	0.046535
4	0.165	0.192264	0.027264
5	0.235	0.244988	0.009988
6	0.215	0.216785	0.001785
7	0.135	0.131540	0.00346
8	0.045	0.052379	0.007379
9	0.01	0.012360	0.00236
	1.0	0.99797	0.046535

- Геометрическое распределение

j	\tilde{w}_j	p_j	$ \tilde{w}_j - p_j $
0	0.490	0.515000	0.025
1	0.280	0.249775	0.030225
2	0.135	0.121141	0.013859
3	0.060	0.058753	0.001247
4	0.020	0.028495	0.008495
5	0.010	0.013820	0.00382
6	0	0.006703	0.006703
7	0.005	0.003251	0.001749
	1.0	0.99694	0.030225

▪ Распределение Пуассона

j	\widetilde{w}_j	p_j	$ \widetilde{w}_j - p_j $
0	0.11	0.104350	0.00565
1	0.2	0.235832	0.035832
2	0.335	0.266490	0.06851
3	0.205	0.200756	0.004244
4	0.085	0.113427	0.028427
5	0.03	0.051269	0.021269
6	0.03	0.019311	0.010689
7	0.005	0.006235	0.001235
	1.0	0.99767	0.06851

4.2 Таблица сравнения рассчитанных характеристик с теоретическими значениями

- Биномиальное распределение

Название показателя	Экспериментальное значение	Теоретическое значение	Абсолютное значение	Относительное отклонение
Выборочное среднее	5.055	5.15	0.095	0.01844
Выборочная дисперсия	2.52198	2.49775	0.02423	0.0097
Выборочное среднее квадратичное отклонение	1.58807	1.58042	0.00765	0.00484
Выборочная мода	5	5	0	0
Выборочная медиана	5	5	0	0
Выборочный коэффициент асимметрии	-0.00781	-0.01898	0.01117	0.01898
Выборочный коэффициент эксцесса	-0.54320	-0.19963	0.34357	0.19963

- Геометрическое распределение

Название показателя	Экспериментальное значение	Теоретическое значение	Абсолютное значение	Относительное отклонение
Выборочное среднее	0.895	0.941747	0.046747	0.049639
Выборочная дисперсия	1.37397	1.82863	0.45466	0.248634
Выборочное среднее квадратичное отклонение	1.1721667	1.35227	0.180103	0.133186
Выборочная мода	0	0	0	---
Выборочная медиана	1	0	1	---
Выборочный коэффициент асимметрии	1.75033	2.132335	0.382005	0.179149
Выборочный коэффициент эксцесса	4.06531	6.546855	2.481545	0.379044

▪ Распределение Пуассона

Название показателя	Экспериментальное значение	Теоретическое значение	Абсолютное значение	Относительное отклонение
Выборочное среднее	2.19	2.26	0.07	0.030973
Выборочная дисперсия	2.023899	2.26	0.236101	0.104469
Выборочное среднее квадратичное отклонение	1.42263	1.50332	0.08069	0.053675
Выборочная мода	2	2	0	0
Выборочная медиана	2	2	0	0
Выборочный коэффициент асимметрии	0.69445	0.66519	0.02926	0.043987
Выборочный коэффициент эксцесса	0.61329	0.44248	0.17081	0.386029

5 Список литературы

- [1] Математическая статистика [Электронный ресурс]: метод. указания по выполнению лаб. работ / А.А. Лобузов — М.: МИРЭА, 2017.
- [2] Боровков А. А. Математическая статистика. — СПб.: Лань, 2010. 704 с.
- [3] Гмурман В.Е. Теория вероятностей и математическая статистика. — М.: Юрайт, 2013. — 479 с.
- [4] Гмурман В.Е. Руководство к решению задач по теории вероятностей и математической статистике. — М.: Юрайт, 2013. — 404 с.
- [5] Емельянов Г.В.Скитович В.П. Задачник по теории вероятностей и математической статистике. — СПб.: Лань, 2007. — 336 с.
- [6] Ивченко Г. И., Медведев Ю. И. Введение в математическую статистику. — М.: Изд-во ЛКИ, 2010. — 599 с.
- [7] Кибзун А.И., Горяинова Е.Р., Наумов А.В. Теория вероятностей и математическая статистика. Базовый курс с примерами и задачам. Учебное пособие — М.:ФИЗМАТЛИТ, 2005. — 232 с.

Приложение

```
import numpy as np
import math
import scipy as sp
import matplotlib.pyplot as plt
import pandas as pd
from decimal import Decimal
#params:
V = 63
EPS = 0.00001
n = 7 + V % 15
p = 0.2 + 0.005*V
N = 200
lamb = 1 + 0.02*V
q = 1 - p

distrub_geom = np.random.geometric(p, N)
- 1
distrub_poisson = np.random.poisson(lamb,
N)
distrub_binom = np.random.binomial(n, p,
N)

def get_frequency_table(xs):
    df = pd.DataFrame({"values" :xs})
    df =
pd.value_counts(df["values"]).to_frame().
reset_index()
    df.columns = ['Value', 'Count']
    df = df.sort_values(by=['Value'])
#Сортировка по значению
    df.reset_index(inplace = True, drop =
True)
    return df

def count_edf(freqs):
    edf = [0]
    s = 0
    for i in freqs:
        s += i
```

```

        edf.append(round(s, 5))
    return edf

def count_k_moment(table, k):
    return sum(table.Frequency *
table.Value**k)

def count_emperical_mean(table):
    return count_k_moment(table, 1)

def count_centra_k_moment(table, k):
    mean = count_emperical_mean(table)
    return sum((table.Value - mean)**k *
table.Frequency)

def plot_edf(edf, values):
    fig = plt.figure(figsize=(15,8))
    ax = fig.add_subplot(111)
    W = 0.005
    n = len(values)
    if (values[0] != 0):
        plt.arrow(values[0] -
1,0,1,0,width=W,head_width = 0.02)

        for i in range(n - 1):
            plt.arrow(values[i], edf[i+1],
values[i+1] - values[i], 0,width=W,
head_width = 0.02)
        ax.set(
            xticks = np.arange(values[0],
values[n - 1] + 1),
            yticks = np.arange(edf[0], edf[-
1] + 0.1, 0.1))

def count_emperical_dispersion(table):
    mean = count_emperical_mean(table)
    sqr = sum(table.Frequency *
table.Value**2)
    return sqr - mean**2

```

```

def count_emperical_std(table):
    return
np.sqrt(count_emperical_dispersion(table)
)

def count_moda(table):
    moda = max(table.Count)
    indx = table.Count[table.Count ==
moda].index.tolist()
    if len(indx) == 1:
        return int(table.Value[indx])
    for j in range(1, len(indx)):
        if abs(indx[j] - indx[j-1]) > 1:
            return "Выборочной моды не
существует"
    return 1/2*(table.Value[indx[0]] +
table.Value[indx[-1]])

def count_skewness(table):
    std = count_emperical_std(table)
    num = count_centra_k_moment(table, 3)
    return num / (std**3)

def count_kurtosis(table):
    std = count_emperical_std(table)
    num = count_centra_k_moment(table, 4)
    return num/(std**4) - 3

def plot_table(table, table_comp):
    fig = plt.figure(figsize=(15,8))
    ax = fig.add_subplot(111)
    plt.plot(table.Value,
table.Frequency, color='b')
    plt.plot(table_comp.Value,
table_comp.P, color="r")
    plt.scatter(table.Value,
table.Frequency, color="black")

```

```

plt.scatter(table_comp.Value,
table_comp.P, color="black")
ax.set(
    xticks = table_comp.Value)

#таблица сравнения относительных частот и
теоретических вероятностей
def create_table_comp(table, type_):
    n = max(table.Value) + 1
    table_comp = pd.DataFrame()
    table_comp['Value'] = np.arange(0, n)

    if type_ == "binom":
        table_comp['P'] = np.zeros(n)
        for i in range(n):
            table_comp.P[i] =
math.comb(n, i)*(p**i)*(q**(n-i))

    if type_ == "geom":
        table_comp['P'] =
(q**table_comp.Value)*p

    if type_ == "poisson":
        table_comp['P'] = np.zeros(n)
        for i in range(n):
            table_comp.P[i] =
(lamb**i)/math.factorial(i)*np.exp(-lamb)

    return table_comp

def main_count(xs, type_):
    mean = 0
    dispersion = 0
    std = 0
    skewness = 0
    kurtosis = 0
    moda = 0
    median = 0

    table = get_frequency_table(xs)

```

```

table['Frequency'] = table.Count / N

edf = count_edf(table.Frequency) #
Empirical distribution function
list[float]
    emp_mean =
count_emperical_mean(table)
    emp_disp =
count_emperical_dispersion(table)
    emp_std = count_emperical_std(table)
    emp_skewness = count_skewness(table)
    emp_kurtosis = count_kurtosis(table)

table_comp = create_table_comp(table,
type_)
plot_table(table, table_comp)
print(table_comp)

if type_ == "binom":
    mean = n*p
    dispersion = n*p*q
    std = np.sqrt(dispersion)
    if int((n+1)*p) - (n+1)*p == 0:
        moda = (n+1)*p - 1/2
    else:
        moda = int((n+1)*p)
    median = int(n*p)
    skewness = (q - p) /
np.sqrt(n*p*q)
    kurtosis = (1 - 6*p*q) / (n*p*q)

if type_ == "poisson":
    mean = lamb
    dispersion = lamb
    std = np.sqrt(lamb)
    moda = int(lamb)
    median = int(lamb + 1/3 -
0.02/lamb)
    skewness = 1/np.sqrt(lamb)
    kurtosis = 1/lamb

if type_ == "geom":

```

```

        mean = q/p
        dispersion = q/p/p
        std = np.sqrt(q)/p
        moda = 0
        if int(np.log(2)/np.log(q)) -
np.log(2)/np.log(q) == 0:
            median = -np.log(2)/np.log(q)
- 1/2
        else:
            median = int(-
np.log(2)/np.log(q))
            skewness = (2-p)/np.sqrt(q)
            kurtosis = 6+p*p/q

    print(table, end='\n\n')
    print("EDF : ", edf, end='\n\n')
    plot_edf(edf, table.Value)

    print("emperical_mean :", emp_mean,
end='\n\n' )
    print("emperical_disp :", emp_disp,
end='\n\n' )
    print("emperical_std  :", emp_std,
end='\n\n' )
    print("emperical moda : ",
count_moda(table), end='\n\n')
    print("emp_skewness  :", emp_skewness,
end='\n\n' )
    print("emp_kurtosis: ", emp_kurtosis,
end='\n\n\n\n')

    print("theoretical characteristics:",
end='\n\n')
    print("mean:", mean, end='\n\n')
    print("disp:", dispersion,
end='\n\n')
    print("std:", std, end='\n\n')
    print("moda: ", moda, end='\n\n')
    print("median: ", median, end='\n\n')
    print("skewness  :", skewness,
end='\n\n' )
    print("kurtosis: ", kurtosis,

```

```
end='\n\n')
```

```
def print_file(xs, type_):  
    with open(type_ + ".txt", 'w') as  
file1:  
        for x in xs:  
            print(x, end='\n',  
file=file1)  
  
        with open("sort" + type_ + ".txt",  
'w') as file2:  
            for x in np.sort(xs):  
                print(x, end='\n',  
file=file2)  
  
print_file(distrub_geom, "geom")  
print_file(distrub_binom, "binom")  
print_file(distrub_poisson, "poisson")  
  
main_count(distrub_geom, "geom")  
main_count(distrub_poisson, "poisson")  
main_count(distrub_binom, "binom")
```