



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

ИНСТИТУТ КИБЕРНЕТИКИ

КАФЕДРА ВЫСШЕЙ МАТЕМАТИКИ

Лабораторная работа 2

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

ВАРИАНТ 63

Тема: **Первичная обработка выборки из**
непрерывной генеральной совокупности

Выполнил:

Студент 3-го курса

Макарчук Н.С

Группа: КМБО-03-18

МОСКВА – 2021

1 Задание

Задание 1. Получить выборку, сгенерировав $N = 200$ псевдослучайных чисел, распределенных по нормальному закону с параметрами $a = -6.3$ и σ^2 , где $\sigma = 1,63$.

Задание 2. Получить выборку, сгенерировав $N = 200$ псевдослучайных чисел, распределенных по показательному закону с параметром $\lambda = 1.37$.

Задание 3. Получить выборку, сгенерировав $N = 200$ псевдослучайных чисел, распределенных равномерно на отрезке $[a, b]$, где $a = -3.15$, $b = 2.85$

Построить:

- 1) группированную выборку (интервальный вариационный ряд) и ассоциированный статистический ряд;
- 2) гистограмму относительных частот;
- 3) график эмпирической функции распределения.

Найти:

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

Составить таблицы:

- 1) сравнения относительных частот и теоретических вероятностей попадания в интервалы;
- 2) сравнения рассчитанных характеристик с теоретическими значениями.

Вычисления проводить с точностью до 0,00001.

2 Краткие теоретические сведения

2.1 Нормальное распределение с параметрами (a, σ^2)

- Плотность

$$f_{\xi}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-a)^2}{2\sigma^2}} = \frac{1}{\sigma} \varphi_0\left(\frac{x-a}{\sigma}\right), \text{ где } \varphi_0(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

- Функция распределения

$$F_{\xi}(x) = \int_{-\infty}^x f_{\xi}(t) dt = \Phi\left(\frac{x-a}{\sigma}\right) \text{ где } \Phi(x) = \int_{-\infty}^x \varphi_0(t) dt$$

- Математическое ожидание

$$M(X) = a$$

- Дисперсия

$$D(X) = \sigma^2$$

- Среднее квадратичное отклонение

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

- Мода

$$M_0 = a$$

- Медиана

$$M_e = a$$

- Коэффициент асимметрии

$$a_s = 0$$

- Коэффициент эксцесса

$$\varepsilon_k = 0$$

2.1 Показательное распределение с параметром $\lambda > 0$

- Плотность

$$f_{\xi}(x) = \begin{cases} 0, & x < 0 \\ \lambda \cdot e^{-\lambda x}, & x \geq 0 \end{cases}$$

- Функция распределения

$$F_{\xi}(x) = \begin{cases} 0, & x \leq 0 \\ \int_0^x \lambda e^{-\lambda t} dt, & x > 0 \end{cases} = \begin{cases} 0, & x \leq 0 \\ 1 - e^{-\lambda x}, & x > 0 \end{cases}$$

- Математическое ожидание

$$M(X) = \lambda^{-1}$$

- Дисперсия

$$D(X) = \lambda^{-2}$$

- Среднее квадратичное отклонение

$$\sigma = \lambda^{-1}$$

- Мода

$$M_0 = 0$$

- Медиана

$$M_e = \frac{\ln(2)}{\lambda}$$

- Коэффициент асимметрии

$$a_s = 2$$

- Коэффициент эксцесса

$$\varepsilon_k = 6$$

2.3 Равномерное непрерывное распределение на [a,b]

- Плотность

$$f_{\xi}(x) = \begin{cases} 0, x \notin (a, b) \\ C, x \in (a, b) \end{cases}, \quad C = \frac{1}{b-a}$$

- Функция распределения

$$F_{\xi}(x) = \int_{-\infty}^x f_{\xi}(t) dt = \begin{cases} 0, x \leq a \\ \frac{x-a}{b-a}, x \in (a, b) \\ 1, x \geq b \end{cases}$$

- Математическое ожидание

$$M(X) = \frac{a+b}{2}$$

- Дисперсия

$$D(X) = \frac{(b-a)^2}{12}$$

- Среднее квадратичное отклонение

$$\sigma = \frac{b-a}{2\sqrt{3}}$$

- Мода

$$M_0 = \frac{a+b}{2}$$

- Медиана

$$M_e = \frac{a+b}{2}$$

- Коэффициент асимметрии

$$a_s = 0$$

- Коэффициент эксцесса

$$\varepsilon_k = -\frac{6}{5} = -1,2$$

2.4 Формулы для экспериментальных расчётов

- Выборочное среднее

$$\bar{x} = \frac{1}{N} \sum_{i=1}^m x_i^* \cdot n_i = \sum_{i=1}^m x_i^* \cdot w_i$$

- Выборочная дисперсия с поправкой Шеппарда

$$s_B^2 = \sum_{i=1}^m (x_i^* - \bar{x})^2 \cdot w_i - \frac{h^2}{12}, \text{ где } h = (a_m - a_0) / m$$

- Выборочное среднее квадратичное отклонение

$$\tilde{\sigma} = \sqrt{s_B^2}$$

- Выборочная мода

$$\bar{M}_0 = a_{k-1} + h \frac{w_k - w_{k-1}}{2w_k - w_{k-1} - w_{k+1}}$$

a_{k-1} – левая граница модального интервала (a_{k-1}, a_k) (интервала, имеющего наибольшую частоту);

w_k – относительная частота на модальном интервале;

w_{k-1}, w_{k+1} – относительные частоты интервалов слева и справа от модального интервала.

- Выборочная медиана

$$\bar{M}_e = a_{k-1} + \frac{h}{w_k} \left(\frac{1}{2} - \sum_{i=1}^{k-1} w_i \right), \text{ если } \sum_{i=1}^{k-1} w_i < \frac{1}{2} < \sum_{i=1}^k w_i ;$$

$$\bar{M}_e = a_k, \text{ если } \sum_{i=1}^k w_i = \frac{1}{2} .$$

- Выборочный момент k-ого порядка

$$\bar{\mu}_k = \overline{x^k} = \sum_{i=1}^m (x_i^*)^k \cdot w_i, \bar{\mu}_1 = \bar{x}$$

- Выборочный центральный момент k-ого порядка

$$\bar{\mu}_k^0 = \sum_{i=1}^m (x_i^* - \bar{x})^k \cdot w_i, \bar{\mu}_1^0 = 0, \bar{\mu}_2^0 = D_B = \bar{\mu}_2 - (\bar{\mu}_1)^2$$

- Выборочный коэффициент асимметрии

$$\bar{\gamma}_1 = \frac{\bar{\mu}_3^0}{\bar{\sigma}^3}$$

- Выборочный коэффициент эксцесса

$$\bar{\gamma}_2 = \frac{\bar{\mu}_4^0}{\bar{\sigma}^4} - 3$$

2.5 Средства языка программирования

В программе расчёта был использован язык программирования Python. Распределения получены следующими функциями модуля `numpy.random`

Нормальное распределение: `numpy.random.normal(a, sigma, N)` с параметрами **a**, **sigma**, и размер выборки **N**

Показательное распределение: `numpy.random.exponential(k, N)` с параметрами где $k = 1/\lambda$ и размер выборки **N**

Равномерное распределение: `numpy.random.uniform(a, b, N)` с параметрами: **a** и **b**— начало и конец отрезка и размер выборки **N**

3 Результаты расчётов

3.1 Задание 1 Нормальное распределение

Параметры : $\mu = -6.3$ $\sigma = 1.63$

▪ Полученная выборка

-8.28246	-4.87334	-4.87341	-4.00788	-6.04288	-7.37175	-6.16747	-5.44755	-7.98567	-7.23263
-8.28246	-4.87334	-4.87341	-4.00788	-6.04288	-7.37175	-6.16747	-5.44755	-7.98567	-7.23263
-5.40644	-7.53543	-7.74495	-4.68845	-7.32521	-5.19338	-5.4559	-5.5711	-7.16705	-3.91663
-5.33579	-5.91668	-5.32458	-8.98652	-6.70636	-8.32404	-5.5619	-5.64462	-6.95233	-6.63279
-9.2616	-4.63674	-5.5687	-6.81252	-5.17414	-8.03433	-6.01637	-3.89349	-3.30026	-3.68309
-4.6912	-6.67763	-7.06163	-4.07369	-5.62135	-8.67886	-5.33811	-6.25834	-8.02797	-10.22378
-8.36385	-5.51977	-5.45891	-7.78993	-7.74842	-4.10856	-7.81704	-5.65683	-6.36093	-4.16158
-7.1145	-5.09225	-5.25859	-4.94325	-6.33795	-6.64645	-7.23267	-9.72492	-7.58918	-3.82536
-6.00458	-6.42226	-5.63142	-6.16176	-3.59162	-7.01636	-7.82244	-4.27608	-7.05571	-8.99672
-7.8198	-7.66973	-8.23246	-4.77318	-5.88115	-6.3282	-7.54397	-8.30249	-6.99154	-7.53139
-4.00225	-6.66934	-7.804	-6.48225	-6.38965	-6.5397	-5.82923	-6.09655	-4.90326	-4.38847
-11.10775	-0.8762	-4.78662	-5.30441	-4.88694	-4.96963	-5.17178	-8.85348	-8.19022	-8.08671
-6.65822	-8.47443	-5.77508	-5.56989	-5.26776	-4.70023	-7.75374	-7.01308	-7.71585	-5.90022
-7.15041	-7.21528	-5.73819	-5.06463	-5.75739	-2.86179	-8.99567	-1.174	-5.27438	-6.75005
-3.7095	-6.90355	-6.85701	-5.44069	-5.07675	-4.09728	-6.97974	-7.85094	-7.89941	-7.19595
-5.8851	-6.51782	-8.1167	-6.16697	-8.61866	-8.44078	-5.73766	-3.8404	-4.95191	-4.36129
-7.03864	-7.07716	-7.05227	-6.12542	-3.87434	-4.10531	-3.96851	-5.06498	-6.31151	-5.08279
-4.91367	-4.76765	-6.4702	-8.20983	-7.40843	-6.33634	-3.771	-8.1054	-6.70433	-7.5827
-5.93368	-4.41309	-8.03807	-5.09468	-5.78152	-5.82133	-6.95932	-3.13325	-6.498	-6.95136
-6.76922	-5.65103	-3.83178	-4.25294	-8.75042	-8.5543	-6.8694	-5.32825	-6.3162	-6.38834
-6.31188	-6.6446	-5.86089	-3.85134	-4.80732	-5.92089	-6.18116	-7.47001	-8.6021	-6.03157

▪ Упорядоченная выборка

-11.10775	-10.22378	-9.72492	-9.2616	-8.99672	-8.99567	-8.98652	-8.85348	-8.75042	-8.67886
-8.61866	-8.6021	-8.5543	-8.47443	-8.44078	-8.36385	-8.32404	-8.30249	-8.28246	-8.23246
-8.20983	-8.19022	-8.1167	-8.1054	-8.08671	-8.03807	-8.03433	-8.02797	-7.98567	-7.89941
-7.85094	-7.82244	-7.8198	-7.81704	-7.804	-7.78993	-7.75374	-7.74842	-7.74495	-7.71585
-7.66973	-7.58918	-7.5827	-7.54397	-7.53543	-7.53139	-7.47001	-7.40843	-7.37175	-7.32521
-7.23267	-7.23263	-7.21528	-7.19595	-7.16705	-7.15041	-7.1145	-7.07716	-7.06163	-7.05571
-7.05227	-7.03864	-7.01636	-7.01308	-6.99154	-6.97974	-6.95932	-6.95233	-6.95136	-6.90355
-6.8694	-6.85701	-6.81252	-6.76922	-6.75005	-6.70636	-6.70433	-6.67763	-6.66934	-6.65822
-6.64645	-6.6446	-6.63279	-6.5397	-6.51782	-6.498	-6.48225	-6.4702	-6.42226	-6.38965
-6.38834	-6.36093	-6.33795	-6.33634	-6.3282	-6.3162	-6.31188	-6.31151	-6.25834	-6.18116
-6.16747	-6.16697	-6.16176	-6.12542	-6.09655	-6.04288	-6.03157	-6.01637	-6.00458	-5.93368
-5.92089	-5.91668	-5.90022	-5.8851	-5.88115	-5.86089	-5.82923	-5.82133	-5.78152	-5.77508
-5.75739	-5.73819	-5.73766	-5.65683	-5.65103	-5.64462	-5.63142	-5.62135	-5.5711	-5.56989
-5.5687	-5.5619	-5.51977	-5.45891	-5.4559	-5.44755	-5.44069	-5.40644	-5.33811	-5.33579
-5.32825	-5.32458	-5.30441	-5.27438	-5.26776	-5.25859	-5.19338	-5.17414	-5.17178	-5.09468
-5.09225	-5.08279	-5.07675	-5.06498	-5.06463	-4.96963	-4.95191	-4.94325	-4.91367	-4.90326
-4.88694	-4.87341	-4.87334	-4.80732	-4.78662	-4.77318	-4.76765	-4.70023	-4.6912	-4.68845
-4.63674	-4.41309	-4.38847	-4.36129	-4.27608	-4.25294	-4.16158	-4.10856	-4.10531	-4.09728
-4.07369	-4.00788	-4.00225	-3.96851	-3.91663	-3.89349	-3.87434	-3.85134	-3.8404	-3.83178
-3.82536	-3.771	-3.7095	-3.68309	-3.59162	-3.30026	-3.13325	-2.86179	-1.174	-0.8762

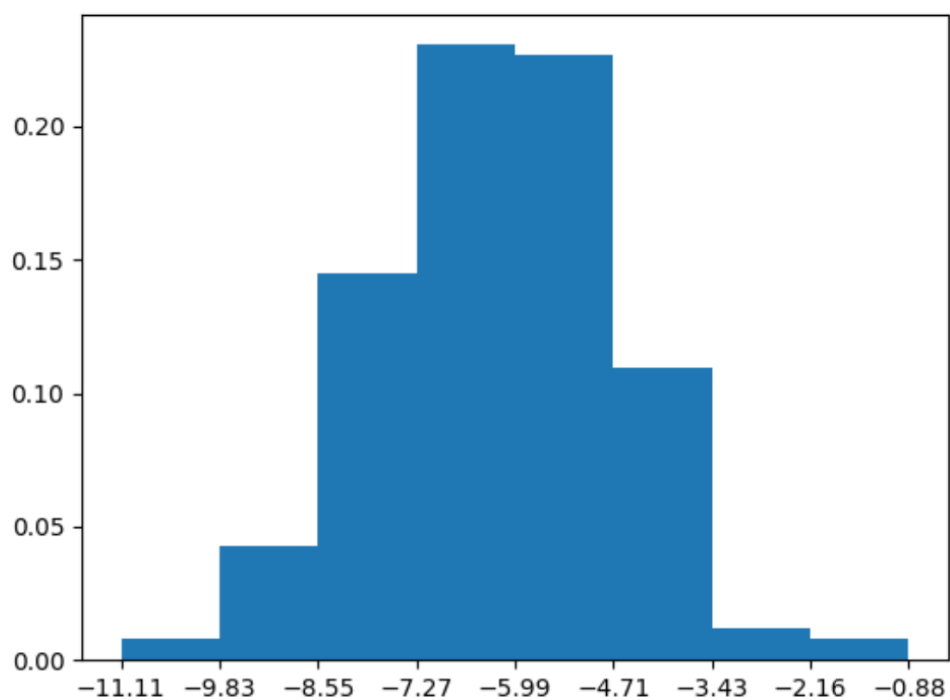
▪ Группированная выборка (интервальный вариационный ряд)

Интервалы	n_i	w_i
[-11.10775, -9.82881]	2	0.010
(-9.82881, -8.54986]	11	0.055
(-8.54986, -7.27092]	37	0.185
(-7.27092, -5.99198]	59	0.295
(-5.99198, -4.71303]	58	0.290
(-4.71303, -3.43409]	28	0.140
(-3.43409, -2.15514]	3	0.015
(-2.15514, -0.8762]	2	0.010
	200	1.0

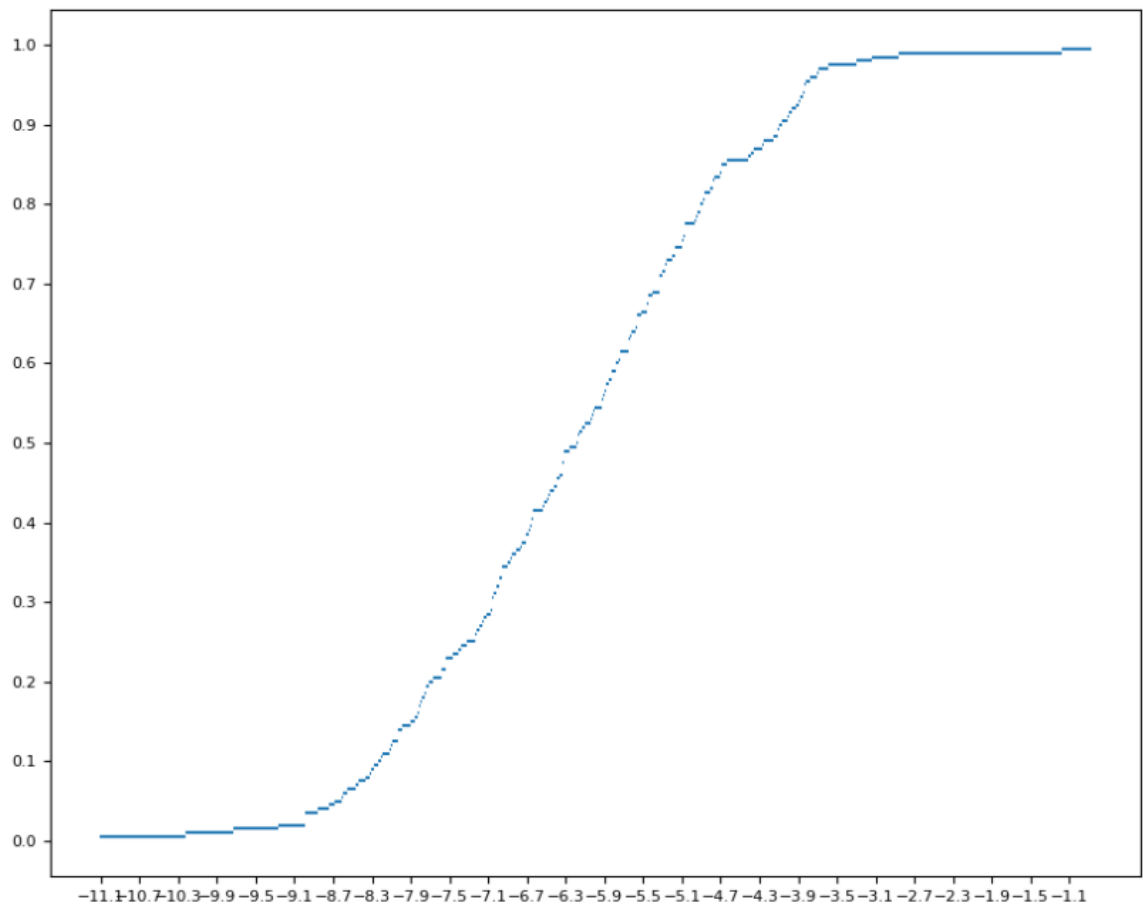
- Ассоциированный статистический ряд

x_i^*	n_i	w_i
-10.46828	2	0.010
-9.18933	11	0.055
-7.91039	37	0.185
-6.63145	59	0.295
-5.35250	58	0.290
-4.07356	28	0.140
-2.79462	3	0.015
-1.51567	2	0.010
	200	1.0

- Гистограмма относительных частот



- График эмпирической функции распределения



- Результаты расчетов требуемых характеристик

Выборочное среднее: -6.2094

Выборочная дисперсия с поправкой Шепарда: 2.39784

Выборочное среднее квадратичное отклонение: 1.5485

Выборочная мода: -6.04758

Выборочная медиана: -6.18707

Выборочный коэффициент асимметрии: 0.00466

Выборочный коэффициент эксцесса: 0.33031

3.2 Задание 2

Показательное распределение

Параметры: $N = 200, \lambda = 1.37$

- Полученная выборка

0.28653	1.38068	0.07761	0.0266	1.42923	0.44247	0.63111	1.04189	0.23131	0.56393
0.21531	1.09143	0.50262	1.00044	0.14418	0.78958	0.68118	0.27496	2.97511	1.71861
0.21237	0.7536	2.02389	0.32224	0.26159	1.53729	1.3936	0.05153	1.73585	0.6084
1.99059	0.35434	2.34194	0.16637	2.27725	0.05454	0.62147	1.06602	0.22582	0.98484
0.56951	0.76163	3.37417	0.2648	0.59385	2.24776	0.48689	0.76547	0.42312	1.32472
0.40856	0.29994	0.06397	0.57864	0.09728	0.19506	0.926	1.49061	0.31942	2.18081
1.15668	0.00352	0.426	0.01124	0.02921	0.11297	0.93822	0.08087	0.04422	0.0398
0.63499	0.42949	0.05195	0.8471	0.29571	0.35039	0.08431	0.08423	0.0951	2.43074
0.1356	0.09442	0.44213	0.39515	2.43318	1.65978	1.12661	3.00905	1.83692	0.77598
0.6061	1.13672	0.35816	0.99298	1.90958	0.72232	0.35706	0.20924	1.28482	0.10012
0.48214	0.62871	1.57335	0.28499	1.06656	0.79057	0.41577	1.056	1.57652	0.31251
0.62078	1.63657	0.08342	0.68361	0.02134	0.84174	1.01223	0.52209	1.92563	0.09955
2.4917	0.73652	1.38839	0.46751	0.04784	0.20636	0.33226	0.3171	0.44201	2.66825
1.3246	2.37721	0.92977	1.41355	0.24963	1.23422	2.05752	0.14439	0.04278	0.239
0.90739	0.2422	1.23787	1.91811	1.42905	0.33976	0.48564	0.60631	0.20216	1.57281
1.31061	0.03609	0.15499	0.07224	0.21636	1.49564	0.37869	0.90661	0.40851	0.04855
2.7038	0.93099	0.17993	1.12987	0.14675	0.67211	0.06573	1.72172	0.53578	0.2997
0.09705	0.21185	0.43576	0.81766	3.1506	0.06301	2.73241	0.24188	0.30142	0.24223
0.07042	0.56247	0.99779	0.58936	2.29636	0.26478	0.07704	0.28464	1.20214	0.30355
0.07043	0.72459	0.04274	0.59455	1.57497	0.76448	0.03088	0.77431	0.68468	0.14526

■ Упорядоченная выборка

0.00352	0.01124	0.02134	0.0266	0.02921	0.03088	0.03609	0.0398	0.04274	0.04278
0.04422	0.04784	0.04855	0.05153	0.05195	0.05454	0.06301	0.06397	0.06573	0.07042
0.07043	0.07224	0.07704	0.07761	0.08087	0.08342	0.08423	0.08431	0.09442	0.0951
0.09705	0.09728	0.09955	0.10012	0.11297	0.1356	0.14418	0.14439	0.14526	0.14675
0.15499	0.16637	0.17993	0.19506	0.20216	0.20636	0.20924	0.21185	0.21237	0.21531
0.21636	0.22582	0.23131	0.239	0.24188	0.2422	0.24223	0.24963	0.26159	0.26478
0.2648	0.27496	0.28464	0.28499	0.28653	0.29571	0.2997	0.29994	0.30142	0.30355
0.31251	0.3171	0.31942	0.32224	0.33226	0.33976	0.35039	0.35434	0.35706	0.35816
0.37869	0.39515	0.40851	0.40856	0.41577	0.42312	0.426	0.42949	0.43576	0.44201
0.44213	0.44247	0.46751	0.48214	0.48564	0.48689	0.50262	0.52209	0.53578	0.56247
0.56393	0.56951	0.57864	0.58936	0.59385	0.59455	0.6061	0.60631	0.6084	0.62078
0.62147	0.62871	0.63111	0.63499	0.67211	0.68118	0.68361	0.68468	0.72232	0.72459
0.73652	0.7536	0.76163	0.76448	0.76547	0.77431	0.77598	0.78958	0.79057	0.81766
0.84174	0.8471	0.90661	0.90739	0.926	0.92977	0.93099	0.93822	0.98484	0.99298
0.99779	1.00044	1.01223	1.04189	1.056	1.06602	1.06656	1.09143	1.12661	1.12987
1.13672	1.15668	1.20214	1.23422	1.23787	1.28482	1.31061	1.3246	1.32472	1.38068
1.38839	1.3936	1.41355	1.42905	1.42923	1.49061	1.49564	1.53729	1.57281	1.57335
1.57497	1.57652	1.63657	1.65978	1.71861	1.72172	1.73585	1.83692	1.90958	1.91811
1.92563	1.99059	2.02389	2.05752	2.18081	2.24776	2.27725	2.29636	2.34194	2.37721
2.43074	2.43318	2.4917	2.66825	2.7038	2.73241	2.97511	3.00905	3.1506	3.37417

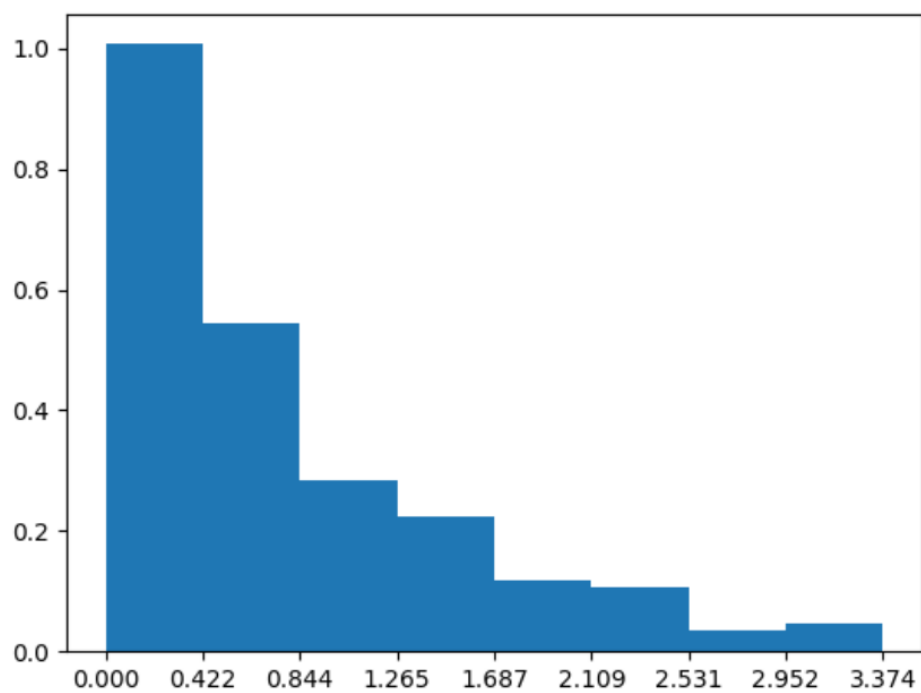
- Группированная выборка (интервальный вариационный ряд)

Интервалы	n_i	w_i
[0.0, 0.42177]	85	0.425
(0.42177, 0.84354]	46	0.230
(0.84354, 1.26531]	24	0.120
(1.26531, 1.68708]	19	0.095
(1.68708, 2.10886]	10	0.050
(2.10886, 2.53063]	9	0.045
(2.53063, 2.9524]	3	0.015
(2.9524, 3.37417]	4	0.020
	200	1.0

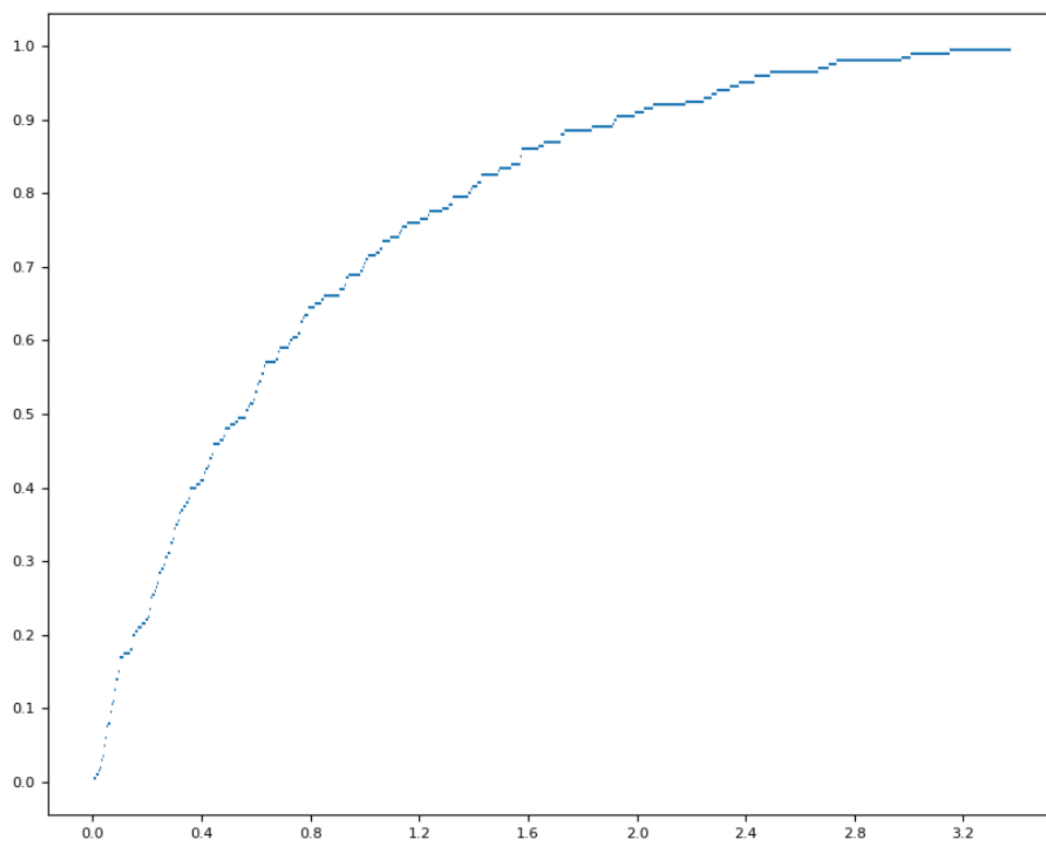
- Ассоциированный статистический ряд

x_i^*	n_i	w_i
0.21089	85	0.425
0.63266	46	0.230
1.05443	24	0.120
1.47620	19	0.095
1.89797	10	0.050
1.89797	9	0.045
2.31974	3	0.015
2.74151	4	0.020
3.16328	200	1.0

- Гистограмма относительных частот



- График эмпирической функции распределения



- Результаты расчетов требуемых характеристик

Выборочное среднее: 0.80559

Выборочная дисперсия с поправкой Шепарда: 0.50245

Выборочное среднее квадратичное отклонение: 0.70884

Выборочная мода: 0.28912

Выборочная медиана: 0.5593

Выборочный коэффициент асимметрии: 1.49445

Выборочный коэффициент эксцесса: 1.79064

3.3 Задание 3

Равномерное распределение

Параметры: $a = -3.15$, $b = 2.85$

- Полученная выборка

0.28653	1.38068	0.07761	0.0266	1.42923	0.44247	0.63111	1.04189	0.23131	0.56393
0.21531	1.09143	0.50262	1.00044	0.14418	0.78958	0.68118	0.27496	2.97511	1.71861
0.21237	0.7536	2.02389	0.32224	0.26159	1.53729	1.3936	0.05153	1.73585	0.6084
1.99059	0.35434	2.34194	0.16637	2.27725	0.05454	0.62147	1.06602	0.22582	0.98484
0.56951	0.76163	3.37417	0.2648	0.59385	2.24776	0.48689	0.76547	0.42312	1.32472
0.40856	0.29994	0.06397	0.57864	0.09728	0.19506	0.926	1.49061	0.31942	2.18081
1.15668	0.00352	0.426	0.01124	0.02921	0.11297	0.93822	0.08087	0.04422	0.0398
0.63499	0.42949	0.05195	0.8471	0.29571	0.35039	0.08431	0.08423	0.0951	2.43074
0.1356	0.09442	0.44213	0.39515	2.43318	1.65978	1.12661	3.00905	1.83692	0.77598
0.6061	1.13672	0.35816	0.99298	1.90958	0.72232	0.35706	0.20924	1.28482	0.10012
0.48214	0.62871	1.57335	0.28499	1.06656	0.79057	0.41577	1.056	1.57652	0.31251
0.62078	1.63657	0.08342	0.68361	0.02134	0.84174	1.01223	0.52209	1.92563	0.09955
2.4917	0.73652	1.38839	0.46751	0.04784	0.20636	0.33226	0.3171	0.44201	2.66825
1.3246	2.37721	0.92977	1.41355	0.24963	1.23422	2.05752	0.14439	0.04278	0.239
0.90739	0.2422	1.23787	1.91811	1.42905	0.33976	0.48564	0.60631	0.20216	1.57281
1.31061	0.03609	0.15499	0.07224	0.21636	1.49564	0.37869	0.90661	0.40851	0.04855
2.7038	0.93099	0.17993	1.12987	0.14675	0.67211	0.06573	1.72172	0.53578	0.2997
0.09705	0.21185	0.43576	0.81766	3.1506	0.06301	2.73241	0.24188	0.30142	0.24223
0.07042	0.56247	0.99779	0.58936	2.29636	0.26478	0.07704	0.28464	1.20214	0.30355
0.07043	0.72459	0.04274	0.59455	1.57497	0.76448	0.03088	0.77431	0.68468	0.14526

▪ Упорядоченная выборка

0.00352	0.01124	0.02134	0.0266	0.02921	0.03088	0.03609	0.0398	0.04274	0.04278
0.04422	0.04784	0.04855	0.05153	0.05195	0.05454	0.06301	0.06397	0.06573	0.07042
0.07043	0.07224	0.07704	0.07761	0.08087	0.08342	0.08423	0.08431	0.09442	0.0951
0.09705	0.09728	0.09955	0.10012	0.11297	0.1356	0.14418	0.14439	0.14526	0.14675
0.15499	0.16637	0.17993	0.19506	0.20216	0.20636	0.20924	0.21185	0.21237	0.21531
0.21636	0.22582	0.23131	0.239	0.24188	0.2422	0.24223	0.24963	0.26159	0.26478
0.2648	0.27496	0.28464	0.28499	0.28653	0.29571	0.2997	0.29994	0.30142	0.30355
0.31251	0.3171	0.31942	0.32224	0.33226	0.33976	0.35039	0.35434	0.35706	0.35816
0.37869	0.39515	0.40851	0.40856	0.41577	0.42312	0.426	0.42949	0.43576	0.44201
0.44213	0.44247	0.46751	0.48214	0.48564	0.48689	0.50262	0.52209	0.53578	0.56247
0.56393	0.56951	0.57864	0.58936	0.59385	0.59455	0.6061	0.60631	0.6084	0.62078
0.62147	0.62871	0.63111	0.63499	0.67211	0.68118	0.68361	0.68468	0.72232	0.72459
0.73652	0.7536	0.76163	0.76448	0.76547	0.77431	0.77598	0.78958	0.79057	0.81766
0.84174	0.8471	0.90661	0.90739	0.926	0.92977	0.93099	0.93822	0.98484	0.99298
0.99779	1.00044	1.01223	1.04189	1.056	1.06602	1.06656	1.09143	1.12661	1.12987
1.13672	1.15668	1.20214	1.23422	1.23787	1.28482	1.31061	1.3246	1.32472	1.38068
1.38839	1.3936	1.41355	1.42905	1.42923	1.49061	1.49564	1.53729	1.57281	1.57335
1.57497	1.57652	1.63657	1.65978	1.71861	1.72172	1.73585	1.83692	1.90958	1.91811
1.92563	1.99059	2.02389	2.05752	2.18081	2.24776	2.27725	2.29636	2.34194	2.37721
2.43074	2.43318	2.4917	2.66825	2.7038	2.73241	2.97511	3.00905	3.1506	3.37417

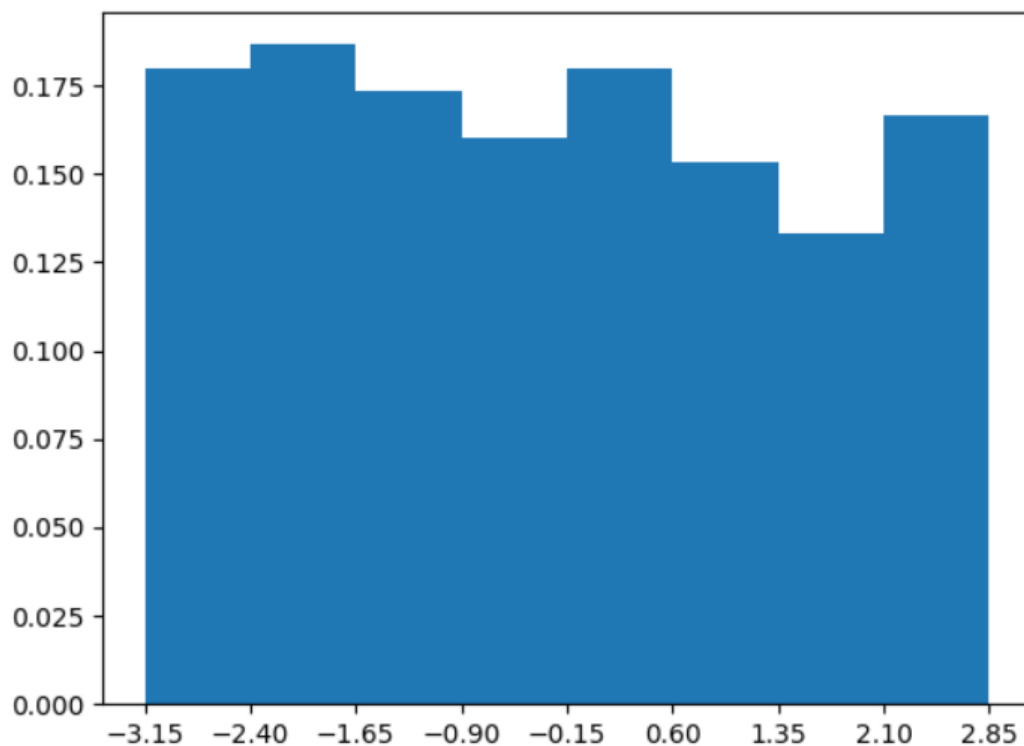
▪ Группированная выборка (интервальный вариационный ряд)

Интервалы	n_i	w_i
[-3.15, -2.4]	27	0.135
(-2.4, -1.65]	28	0.140
(-1.65, -0.9]	26	0.130
(-0.9, -0.15]	24	0.120
(-0.15, 0.6]	27	0.135
(0.6, 1.35]	23	0.115
(1.35, 2.1]	20	0.100
(2.1, 2.85]	25	0.125
	200	1.0

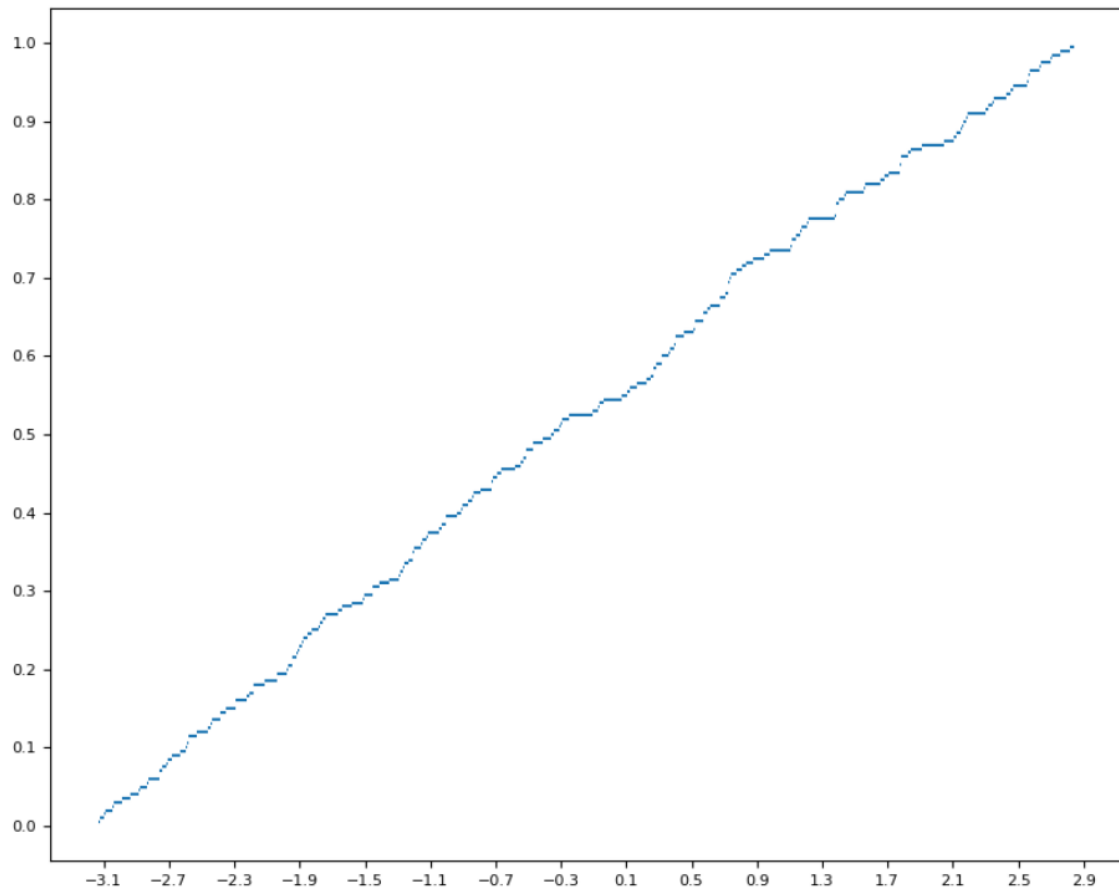
- Ассоциированный статистический ряд

x_i^*	n_i	w_i
-2.775	27	0.135
-2.025	28	0.140
-1.275	26	0.130
-0.525	24	0.120
0.225	27	0.135
0.975	23	0.115
1.725	20	0.100
2.475	25	0.125
	200	1.0

- Гистограмма относительных частот



- График эмпирической функции распределения



- Результаты расчетов требуемых характеристик

Выборочное среднее: -0.2625

Выборочная дисперсия с поправкой Шепарда: 2.9064

Выборочное среднее квадратичное отклонение: 1.70482

Выборочная мода: -2.15

Выборочная медиана: -0.30625

Выборочный коэффициент асимметрии: 0.10862

Выборочный коэффициент эксцесса: -1.13838

4 Анализ результатов и выводы

4.1 Таблица сравнения относительных частот и теоретических вероятностей

- Нормальное распределение с параметрами (a, σ^2)

Интервалы	w_i	p_i	$ w_i - p_i $
$[-11.10775, -9.82881]$	0.010	0.01361	0.00361
$(-9.82881, -8.54986]$	0.055	0.06855	0.01355
$(-8.54986, -7.27092]$	0.185	0.19195	0.00695
$(-7.27092, -5.99198]$	0.295	0.29924	0.00424
$(-5.99198, -4.71303]$	0.290	0.25993	0.03007
$(-4.71303, -3.43409]$	0.140	0.12577	0.01423
$(-3.43409, -2.15514]$	0.015	0.03386	0.01886
$(-2.15514, -0.8762]$	0.010	0.00506	0.00494
	1.0	0.99797	0.03007

- Показательное распределение с параметром $\lambda > 0$

Интервалы	w_i	p_i	$ w_i - p_i $
$[0.0, 0.42177]$	0.425	0.43888	0.01388
$(0.42177, 0.84354]$	0.230	0.24626	0.01626
$(0.84354, 1.26531]$	0.120	0.13818	0.01818
$(1.26531, 1.68708]$	0.095	0.07754	0.01746
$(1.68708, 2.10886]$	0.050	0.04351	0.00649
$(2.10886, 2.53063]$	0.045	0.02441	0.02059
$(2.53063, 2.9524]$	0.015	0.01370	0.00130
$(2.9524, 3.37417]$	0.020	0.00769	0.01231
	1.0	0.99017	0.02059

- Равномерное непрерывное распределение на $[a, b]$

Интервалы	w_i	p_i	$ w_i - p_i $
$[-3.15, -2.4]$	0.135	0.125	0.010
$(-2.4, -1.65]$	0.140	0.125	0.015
$(-1.65, -0.9]$	0.130	0.125	0.005
$(-0.9, -0.15]$	0.120	0.125	0.005
$(-0.15, 0.6]$	0.135	0.125	0.010
$(0.6, 1.35]$	0.115	0.125	0.010
$(1.35, 2.1]$	0.100	0.125	0.025
$(2.1, 2.85]$	0.125	0.125	0.000
	1.0	1.0	0.025

4.2 Таблица сравнения рассчитанных характеристик с теоретическими значениями

- Нормальное распределение с параметрами (α, σ^2)

Название показателя	Экспериментальное значение	Теоретическое значение	Абсолютное отклонение	Относительное отклонение
Выборочное среднее	-6.20940	-6.3000	0.09060	0.01438
Выборочная дисперсия с поправкой Шеппарда	2.39784	2.6569	0.25906	0.09750
Выборочное среднее квадратичное отклонение	1.54850	1.6300	0.08150	0.05000
Выборочная мода	-6.04758	-6.3000	0.25242	0.04007
Выборочная медиана	-6.18707	-6.3000	0.11293	0.01793
Выборочный коэффициент асимметрии	0.00466	0.0	0.00466	-
Выборочный коэффициент эксцесса	0.33031	0.0	0.33031	-

- Показательное распределение с параметром $\lambda > 0$

Название показателя	Экспериментальное значение	Теоретическое значение	Абсолютное отклонение	Относительное отклонение
Выборочное среднее	0.80559	0.72993	0.07566	0.10365
Выборочная дисперсия с поправкой Шеппарда	0.50245	0.53279	0.03034	0.05695
Выборочное среднее квадратичное отклонение	0.70884	0.72993	0.02109	0.02889
Выборочная мода	0.28912	0.0	0.28912	-
Выборочная медиана	0.55930	0.50595	0.05335	0.10545
Выборочный коэффициент асимметрии	1.49445	2.0	0.50555	0.25278
Выборочный коэффициент эксцесса	1.79064	6.0	4.20936	0.70156

- Равномерное непрерывное распределение на [a,b]

Название показателя	Экспериментальное значение	Теоретическое значение	Абсолютное отклонение	Относительное отклонение
Выборочное среднее	-0.26250	-0.15000	0.11250	0.75000
Выборочная дисперсия с поправкой Шепарда	2.90640	3.00000	0.09360	0.03120
Выборочное среднее квадратичное отклонение	1.70482	1.73205	0.02723	0.01572
Выборочная мода	-2.15000	-0.15000	2.00000	13.33333
Выборочная медиана	-0.30625	-0.15000	0.15625	1.04167
Выборочный коэффициент асимметрии	0.10862	0.00000	0.10862	-
Выборочный коэффициент эксцесса	-1.13838	-1.20000	0.06162	0.05135

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 params as prm
import cont_distributions as dstrb
import matplotlib.pyplot as plt

if __name__ == "__main__":
    uniform = dstrb.Uniform(prm.segment[0],
prm.segment[1])
    normal = dstrb.Normal(prm.a, prm.sigma)
    exp = dstrb.Exponential(prm.lamb)

    xs_uniform =
np.loadtxt("uniform_distribution.txt", dtype=float)
    xs_normal = np.loadtxt("normal_distribution.txt",
dtype=float)
    xs_exp =
np.loadtxt("exponential_distribution.txt",
dtype=float)

    emp_1 = dstrb.Empirical(xs_uniform, "Uniform")
    emp_2 = dstrb.Empirical(xs_normal, "Normal")
    emp_3 = dstrb.Empirical(xs_exp, "Exponential")

    for emp in [(emp_1, uniform), (emp_2, normal),
(emp_3, exp)]:
        print("type = ", emp[1].type, end='\n')
        emp[0].show()
        emp[0].plot_hist()
        emp[0].plot_edf()
        emp[0].print_interval_table()
        prop_error_table =
dstrb.create_prop_error_table(emp[0], emp[1])
        print(prop_error_table, end='\n')
        emp[0].plot_hist()
        emp[0].plot_edf()
        interval_error_table =
dstrb.create_interval_error_table(emp[0], emp[1])
        print(interval_error_table)
        print("-----")

import params as prm
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import collections as mc
import math

class Base:
    def __init__(self):
```

```

self.type = 0
self.mean = 0
self.dispersion = 0
self.std = 0
self.mode = 0
self.median = 0
self.skewness = 0 # коэф асимметрии
self.kurtosis = 0 # коэф эксцесса

def show(self):
    print("Properties of ", self.type, end='\n')
    print("Выборочное среднее: ",
round(self.mean, 5), end='\n')
    print("Выборочная дисперсия с поправкой
Шепарда: ", round(self.dispersion, 5), end='\n')
    print("Выборочное среднее квадратичное
отклонение: ", round(self.std, 5), end='\n')
    print("Выборочная мода: ", round(self.mode,
5), end='\n')
    print("Выборочная медиана: ",
round(self.median, 5), end='\n')
    print("Выборочный коэффициент ассиметрии:",
round(self.skewness, 5), end='\n')
    print("Выборочный коэффициент эксцесса: ",
round(self.kurtosis, 5), end='\n\n')

def get_list_props(self):
    return [round(self.mean,5),
            round(self.dispersion,5),
            round(self.std,5),
            round(self.mode,5),
            round(self.median,5),
            round(self.skewness,5),
            round(self.kurtosis,5)]

def CDF(self, x): # Cumulative distribution
function
    return 0

def count_prob_interval(self, interval:
pd.Interval):
    return self.CDF(interval.right) -
self.CDF(interval.left)

class Uniform(Base):
    def __init__(self, a, b):
        super().__init__()
        self.a = a
        self.b = b
        self.type = "Uniform"
        self.mean = (a + b) / 2
        self.dispersion = (b - a) ** 2 / 12
        self.std = (b - a) / 2 / np.sqrt(3)

```

```

        self.mode = self.mean
        self.median = self.mean
        self.skewness = 0
        self.kurtosis = -6 / 5

    def CDF(self, x):
        if x < self.a:
            return 0
        if x >= self.b:
            return 1
        return (x - self.a) / (self.b - self.a)

class Normal(Base):
    def __init__(self, a, sigma):
        super().__init__()
        self.type = "Normal"
        self.mean = a
        self.dispersion = sigma ** 2
        self.std = sigma
        self.mode = a
        self.median = a
        self.skewness = 0
        self.kurtosis = 0
        self.a = a
        self.sigma = sigma

    def CDF(self, x):
        return 0.5 * (1 + math.erf((x - self.a) /
math.sqrt(2) / self.sigma))

class Exponential(Base):
    def __init__(self, lamb):
        super().__init__()
        self.type = "Exponential"
        self.mean = 1 / lamb
        self.dispersion = 1 / lamb / lamb
        self.std = self.mean
        self.mode = 0
        self.median = np.log(2) / lamb
        self.skewness = 2
        self.kurtosis = 6
        self.lamb = lamb

    def CDF(self, x):
        if x < 0:
            return 0
        else:
            return 1 - np.exp(-self.lamb * x)

def create_prop_error_table(empiric, theoretical):
    struct = {"Empirical": empiric.get_list_props(),

```

```

        "Theoretical":
theoretical.get_list_props()}
    prop_error_table = pd.DataFrame(struct)
    prop_error_table.index = ["mean",
                              "dispersion",
                              "std",
                              "mode",
                              "median",
                              "skewness",
                              "kurtosis"]
    prop_error_table["Absolute_error"] =
round(np.abs(prop_error_table["Empirical"] -

prop_error_table["Theoretical"]), 5)
    prop_error_table["Relative_error"] =
round(np.abs(prop_error_table["Absolute_error"] \
/
prop_error_table["Theoretical"]), 5)
    return prop_error_table

def create_interval_error_table(emp, theory):
    struct = {"Intervals":emp.df["Intervals"],
"Frequency":emp.df["Frequency"]}
    table = pd.DataFrame(struct)
    probes = [round(theory.count_prob_interval(seg),
5) for seg in table["Intervals"]]
    table["theory_prob"] = probes
    table["error"] = np.abs(table["Frequency"] -
table["theory_prob"])
    print("sum : ", sum(table["theory_prob"]),
end='\n')
    return table

class Empirical(Base):
    def __init__(self, data, type_="Unknown"):
        super().__init__()
        self.type = type_
        self.__size = len(data)
        self.__data = np.sort(data)
        self.__create_bins()
        self.df = self.__create_interval_df() # main
interval dataframe
        self.__add_midpoint()
        # counting:
        self.__count_mean()
        self.__count_dispersion() # with Sheppard's
correction
        self.__count_std()
        self.__count_mode()
        self.__count_median()
        self.__count_skewness()
        self.__count_kurtosis()

```

```

    # self.test()

    def plot_hist(self):
        fig, ax = plt.subplots()
        plt.hist(self.__data, self.__bins,
density=True)
        ax.set_xticks(self.__bins)
        ax.autoscale()
        fig.savefig("hist_" + self.type + '.png')

    def plot_edf(self):
        fig, ax = plt.subplots(figsize=(10, 8))
        n = self.__size
        lines = [(self.__data[i], (i + 1) / n),
(self.__data[i + 1], (i + 1) / n)] for i in range(n -
1)]
        lc = mc.LineCollection(lines)
        ax.add_collection(lc)
        ax.autoscale()
        ax.set_yticks(np.arange(0, 1.1, 0.1))
        ax.set_xticks(np.arange(round(self.__data[0],
1), round(self.__data[-1], 1) + 0.2, 0.4))
        ax.tick_params(labelsize=8)
        # ax.margins(0.1)
        fig.savefig("edf_" + self.type + '.png')

    def __create_bins(self):
        lower = self.__data[0]
        upper = self.__data[-1]
        if self.type == "Exponential":
            lower = 0
        if self.type == "Uniform":
            lower = prm.segment[0]
            upper = prm.segment[1]

        d = upper - lower
        m = prm.count_intervals
        h = d / m
        self.__bins = np.zeros(m + 1)
        self.__bins[0] = lower
        self.__bins[-1] = upper
        for i in range(1, len(self.__bins) - 1):
            self.__bins[i] = self.__bins[i - 1] + h

    def __add_midpoint(self):
        xs = [round((self.__bins[i] + self.__bins[i -
1]) / 2, 5)
            for i in range(1, len(self.__bins))] #
Midpoint of the segment
        self.df["Midpoint"] = xs

    def __create_interval_df(self):
        s = pd.Series(self.__data)
        cut = pd.cut(s, bins=self.__bins, right=True,

```

```

include_lowest=True, precision=5)
    res = s.groupby(cut).size()
    intervals = res.index.tolist()
    intervals[0] = pd.Interval(self.__bins[0],
intervals[0].right, closed='both')
    df = pd.DataFrame({"Intervals": intervals,
"Count": res.tolist()})
    df["Frequency"] = df["Count"] / self.__size
    return df

    def __count_centre_k_moment(self, k):
        return sum((self.df.Midpoint - self.mean) **
k * self.df.Frequency)

    def __count_k_moment(self, k):
        return sum((self.df.Midpoint ** k) *
self.df.Frequency)

    def __count_mean(self):
        self.mean = self.__count_k_moment(1)

    def __count_dispersion(self, corr=True):
        self.dispersion =
self.__count_centre_k_moment(2)
        if corr:
            self.dispersion -= 1 / 12 *
(self.__data[-1] - self.__data[0]) /
prm.count_intervals

    def __count_std(self):
        self.std = np.sqrt(self.dispersion)

    def __count_median(self):
        h = (self.__bins[-1] - self.__bins[0]) /
prm.count_intervals
        s = self.df["Frequency"][0]
        k = 0
        while s <= 1 / 2:
            k += 1
            s += self.df["Frequency"][k]
        w_k = self.df["Frequency"][k]
        left = self.df["Intervals"][k].left
        right = self.df["Intervals"][k].right
        if s == 1 / 2:
            self.median = right
        else:
            self.median = left + (h / w_k) * (0.5 -
(s - w_k))

    def __count_mode(self):
        idx_max = self.df["Count"].idxmax()
        max_freq = self.df["Frequency"][idx_max]
        right_freq, left_freq = 0, 0
        if idx_max > 0:

```



```

        left_freq = self.df["Frequency"][idx_max
- 1]
        if idx_max + 1 < prm.count_intervals:
            right_freq = self.df["Frequency"][idx_max
+ 1]
        h = (self.__bins[-1] - self.__bins[0]) /
prm.count_intervals
        left_mode =
self.df["Intervals"][idx_max].left
        self.mode = left_mode + h * ((max_freq -
left_freq) / (2 * max_freq - left_freq - right_freq))

    def __count_skewness(self):
        self.skewness =
self.__count_centre_k_moment(3) / (self.std ** 3)

    def __count_kurtosis(self):
        self.kurtosis =
self.__count_centre_k_moment(4) / (self.std ** 4) - 3

    def print_interval_table(self):
        print(self.df, end='\n\n')

```