



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 67

Тема: _____ «Первичная обработка выборки из
_____ непрерывной генеральной совокупности» _____

Выполнил:
Студент 3-го курса
Мусатов Д. Ю

Группа: КМБО-03-18

МОСКВА – 2021

Содержание

1	Задание	3
2	Краткие теоретические сведения	4
2.1	Нормальное распределение	4
2.2	Показательное распределение	5
2.3	Равномерное распределение	6
2.4	Формулы для расчёта выборочных значений	7
3	Результаты расчётов	8
3.1	Задание №1	8
3.2	Задание №2	12
3.3	Задание №3	16
4	Анализ результатов и выводы	20
5	Список использованной литературы	22

1 Задание

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

$$a = (-1)^V \cdot 0,01 \cdot V; \quad \sigma = 0,01 \cdot V + 1$$

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

$$\lambda = 2 + (-1)^V \cdot 0,01 \cdot V$$

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

$$a = (-1)^V \cdot 0,05 \cdot V \quad b = a + 6$$

V – номер варианта.

Для каждого Задания:

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 Шепарда;
- 3) выборочное среднее квадратическое 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

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

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

Математическое ожидание: $M(x) = a$

Дисперсия: $D(x) = \sigma^2$

Среднее квадратическое отклонение: $\sigma = \sigma$

Мода: $M_0 = a$

Медиана: $M_e = a$

Коэффициент асимметрии: $a_s = 0$

Коэффициент эксцесса: $\varepsilon_k = 0$

2.2 Показательное распределение

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

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

Математическое ожидание:
$$M(x) = \frac{1}{\lambda}$$

Дисперсия:
$$D(x) = \frac{1}{\lambda^2}$$

Среднее квадратическое отклонение:
$$\sigma = \sqrt{D(x)} = \frac{1}{\lambda}$$

Мода:
$$M_0 = 0$$

Медиана:
$$M_e = \frac{\ln 2}{\lambda}$$

Коэффициент асимметрии:
$$a_s = 2$$

Коэффициент эксцесса:
$$\varepsilon_k = 6$$

2.3 Равномерное распределение

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

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:
$$M = \frac{a+b}{2}$$

Дисперсия:
$$D = \frac{(b-a)^2}{12}$$

Среднее квадратическое отклонение:
$$\sigma = \sqrt{D} = \frac{b-a}{2\sqrt{3}}$$

Мода:
$$M_0 = \frac{a+b}{2}$$

Медиана:
$$M_e = \frac{a+b}{2}$$

Коэффициент асимметрии:
$$a_s = 0$$

Коэффициент эксцесса:
$$\varepsilon_k = -\frac{6}{5}$$

2.4 Формулы для расчёта выборочных значений

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 Шепарда:
$$s_B^2 = \sum_{i=1}^m (x_i^* - \bar{x})^2 \cdot w_i - \frac{h^2}{12},$$

где $h = \frac{(a_m - a_0)}{m}$, $m = 1 + [\log_2 N]$

Выборочное среднее квадратическое отклонение:
$$\bar{\sigma} = \sqrt{s_B^2}$$

Выборочная мода:
$$\overline{M}_0 = a_{k-1} + h \cdot \frac{w_k - w_{k-1}}{2 \cdot w_k - w_{k-1} - w_{k+1}}$$

Выборочная медиана:
$$\overline{M}_e = a_{k-1} + \frac{h}{w_k} \left(\frac{1}{2} - \sum_{i=1}^{k-1} w_i \right)$$

Выборочный центральный момент k -ого порядка:
$$\overline{\mu}_k^0 \sum_{i=1}^m (x_i^* - \bar{x})^k \cdot w_i$$

Выборочный коэффициент асимметрии:
$$\bar{\gamma}_1 = \frac{\overline{\mu}_3^0}{\overline{\sigma}^3}$$

Выборочный коэффициент эксцесса:
$$\bar{\gamma}_2 = \frac{\overline{\mu}_4^0}{\overline{\sigma}^4} - 3$$

a_{k-1} — левая граница интервала

w_k — относительная частота на интервале;

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

3 Результаты расчётов

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

3.1 Задание №1

$$a = -0.67 \quad \sigma = 1.67 \quad N = 200$$

Полученная выборка

1,68816	0,69207	-2,51545	-0,47829	-5,42243	-3,65343	-1,92482	-3,22744	0,13797	-0,34816
-1,17941	1,77815	-0,18332	-1,41109	-1,11171	-1,65243	0,08095	1,80188	-4,38515	-0,67354
-0,50975	1,42218	-2,33087	2,46099	-1,23931	-0,0242	-1,62983	-2,52144	3,64422	-0,99817
-1,46121	-0,95339	-2,22653	-1,11588	-3,47623	-1,08225	-0,6821	-2,37548	-1,55195	0,87922
0,70785	-0,81458	-3,49707	2,01116	-0,5269	-1,96667	3,75455	0,84639	-3,10826	-1,6396
1,74899	-2,0251	-0,89795	-0,39182	-1,4293	0,58158	-1,9631	0,92285	1,99788	1,13817
0,1955	-1,25485	1,66725	0,04514	-0,21607	0,47774	-0,90101	0,60861	0,72966	-2,44785
-0,62456	-5,07347	0,77834	-0,10291	0,76366	4,25706	-1,04013	-3,13953	-0,12163	-2,74869
1,783	-2,53312	0,22286	1,62201	-0,29144	0,35133	0,65559	-2,15866	-0,04121	-1,20789
0,01306	-2,97574	-1,48045	1,96936	-1,4917	-1,19325	-0,99262	0,75705	-1,91101	-1,19849
-1,18935	-2,41354	1,59787	0,64976	0,75722	-2,61799	1,21498	-0,92585	0,3861	-0,92726
0,98492	1,83366	-0,74896	-0,13868	2,38383	-1,48891	0,95151	-0,90006	0,15432	0,24988
-0,46324	-4,45304	-0,32454	0,05355	1,65001	0,23673	-2,42996	-2,32779	-0,24034	-0,32113
-2,67836	0,0907	1,17742	-3,37488	-2,53553	-2,52476	1,5883	-1,02026	-0,96268	0,63484
1,75233	2,14343	1,71463	2,3553	0,67605	-0,39467	3,52145	0,29118	-2,7384	-1,30685
-0,08288	2,47532	1,45777	-2,51103	-1,24931	-5,00819	1,52342	-0,48768	-3,17531	-3,48973
-0,53879	0,69144	-1,05351	-3,05545	2,20225	-2,11609	0,03139	0,99297	-1,06437	-0,73306
1,38796	-1,45143	0,68883	-0,52791	-5,39227	-1,14608	-0,65385	-0,82493	-0,95897	0,04163
-0,12146	-0,63389	-3,42178	-1,00181	0,72701	1,65146	-0,06283	0,54828	-1,83798	0,55846
-1,24385	-2,23097	1,42247	-1,84705	-2,61592	-2,24439	0,05825	-0,93236	-3,04376	-0,70548

Упорядоченная выборка

-5,42243	-5,39227	-5,07347	-5,00819	-4,45304	-4,38515	-3,65343	-3,49707	-3,48973	-3,47623
-3,42178	-3,37488	-3,22744	-3,17531	-3,13953	-3,10826	-3,05545	-3,04376	-2,97574	-2,74869
-2,7384	-2,67836	-2,61799	-2,61592	-2,53553	-2,53312	-2,52476	-2,52144	-2,51545	-2,51103
-2,44785	-2,42996	-2,41354	-2,37548	-2,33087	-2,32779	-2,24439	-2,23097	-2,22653	-2,15866
-2,11609	-2,0251	-1,96667	-1,9631	-1,92482	-1,91101	-1,84705	-1,83798	-1,65243	-1,6396
-1,62983	-1,55195	-1,4917	-1,48891	-1,48045	-1,46121	-1,45143	-1,4293	-1,41109	-1,30685
-1,25485	-1,24931	-1,24385	-1,23931	-1,20789	-1,19849	-1,19325	-1,18935	-1,17941	-1,14608
-1,11588	-1,11171	-1,08225	-1,06437	-1,05351	-1,04013	-1,02026	-1,00181	-0,99817	-0,99262
-0,96268	-0,95897	-0,95339	-0,93236	-0,92726	-0,92585	-0,90101	-0,90006	-0,89795	-0,82493
-0,81458	-0,74896	-0,73306	-0,70548	-0,6821	-0,67354	-0,65385	-0,63389	-0,62456	-0,53879
-0,52791	-0,5269	-0,50975	-0,48768	-0,47829	-0,46324	-0,39467	-0,39182	-0,34816	-0,32454
-0,32113	-0,29144	-0,24034	-0,21607	-0,18332	-0,13868	-0,12163	-0,12146	-0,10291	-0,08288
-0,06283	-0,04121	-0,0242	0,01306	0,03139	0,04163	0,04514	0,05355	0,05825	0,08095
0,0907	0,13797	0,15432	0,1955	0,22286	0,23673	0,24988	0,29118	0,35133	0,3861
0,47774	0,54828	0,55846	0,58158	0,60861	0,63484	0,64976	0,65559	0,67605	0,68883
0,69144	0,69207	0,70785	0,72701	0,72966	0,75705	0,75722	0,76366	0,77834	0,84639
0,87922	0,92285	0,95151	0,98492	0,99297	1,13817	1,17742	1,21498	1,38796	1,42218
1,42247	1,45777	1,52342	1,5883	1,59787	1,62201	1,65001	1,65146	1,66725	1,68816
1,71463	1,74899	1,75233	1,77815	1,783	1,80188	1,83366	1,96936	1,99788	2,01116
2,14343	2,20225	2,3553	2,38383	2,46099	2,47532	3,52145	3,64422	3,75455	4,25706

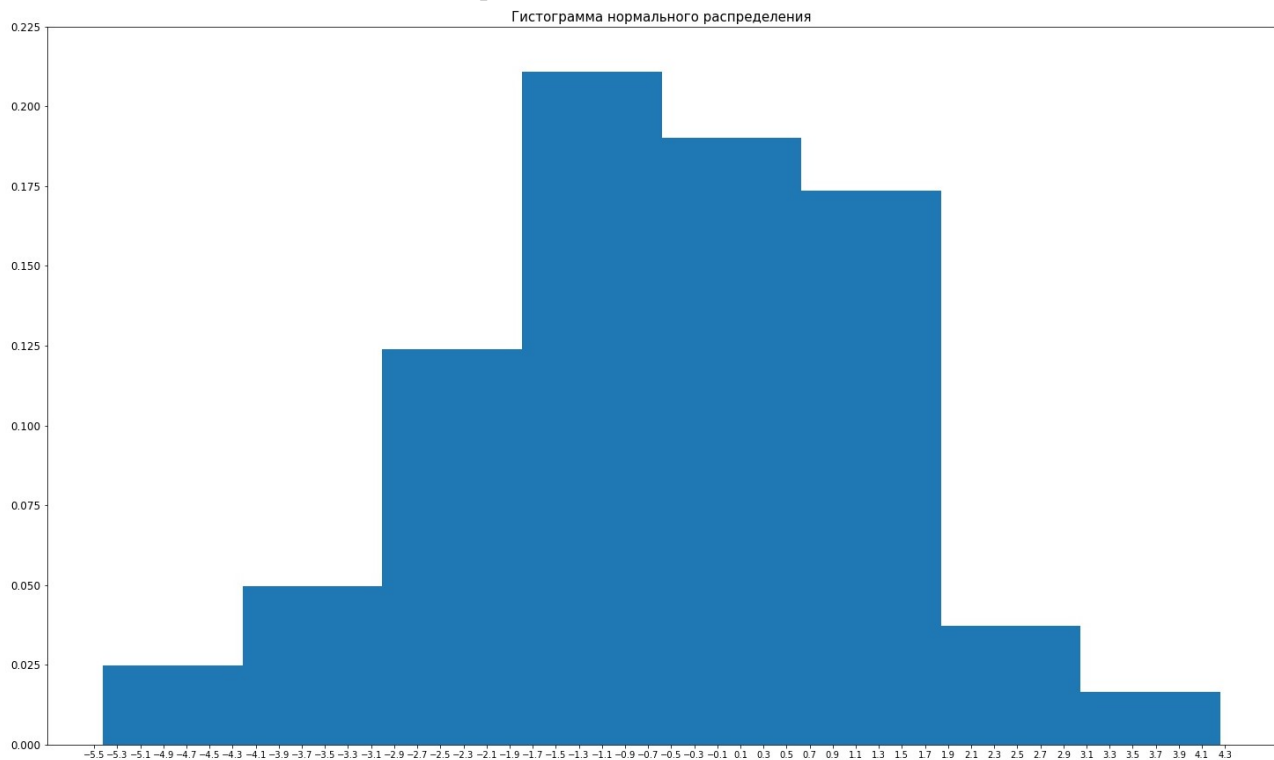
Группированная выборка(интервальный вариационный ряд)

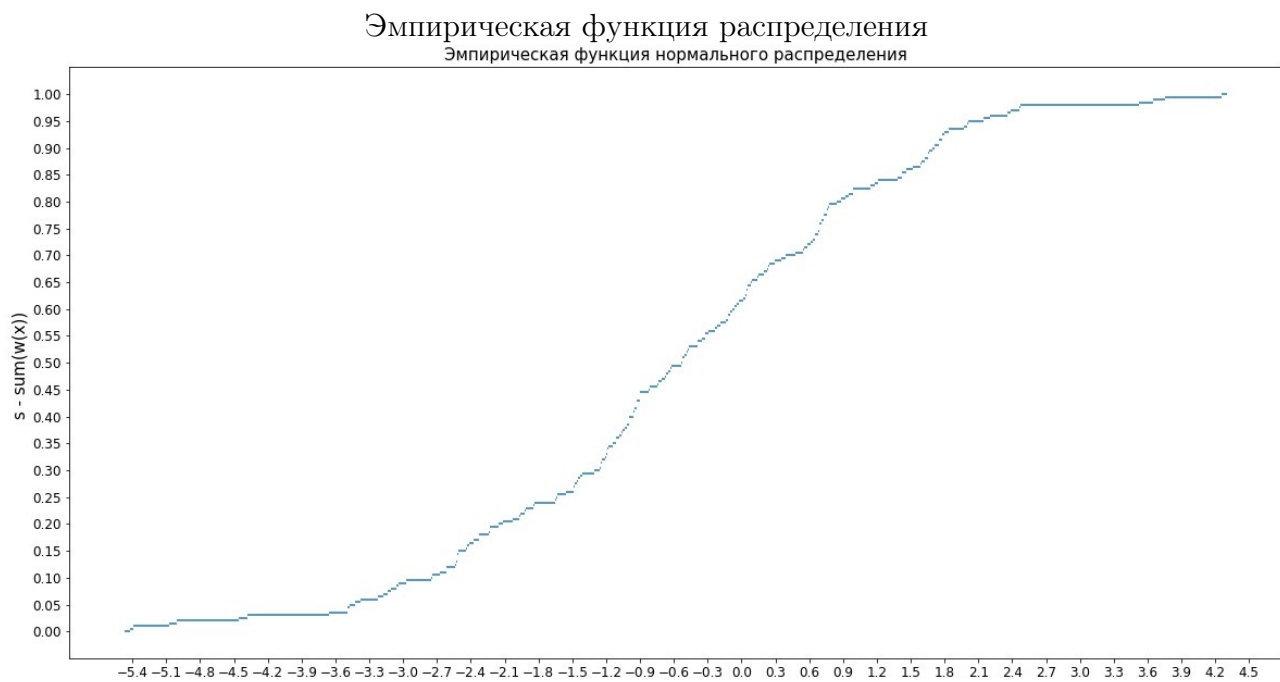
Интервалы	n_j	w_j
$[-5.42243, -4.2125]$	6	0.03
$(-4.2125, -3.00256]$	12	0.06
$(-3.00256, -1.79262]$	30	0.15
$(-1.79262, -0.58269]$	51	0.255
$(-0.58269, 0.62725]$	46	0.23
$(0.62725, 1.83718]$	42	0.21
$(1.83718, 3.04712]$	9	0.045
$(3.04712, 4.25706]$	4	0.02
	200	1

Ассоциированный статистический ряд

x^*_j	n'_j	w'_j
-4.81746	6	0.03
-3.60753	12	0.06
-2.39759	30	0.15
-1.18766	51	0.255
0.02228	46	0.23
1.23222	42	0.21
2.44215	9	0.045
3.65209	4	0.02
	200	1

Гистограмма относительных частот





- Выборочное среднее: -0.57664;
- Выборочная дисперсия с поправкой Шепарда: 3.09865;
- Выборочное среднее квадратическое отклонение: 1.7603;
- Выборочная мода: -0.81537;
- Выборочная медиана: -0.55639;
- Выборочный коэффициент асимметрии: -0.16131;
- Выборочный коэффициент эксцесса: 0.00593;

3.2 Задание №2

$$\lambda = 1.33 \quad N = 200$$

Полученная выборка

0,02491	0,1714	0,27751	0,48437	0,39415	1,16019	0,25626	1,97856	0,08407	1,89257
0,53269	3,36295	0,89313	0,70238	0,83937	0,38153	0,52902	0,54266	0,10275	1,43742
1,61255	1,05899	1,50939	1,05011	0,67258	1,20179	1,99197	0,79823	2,00839	0,52722
0,21905	0,87149	0,02226	0,66328	0,49385	0,52196	3,62593	0,03376	0,11753	0,61679
0,08173	0,45393	1,09834	0,26439	1,0284	0,75013	0,63648	3,2017	1,01997	0,10856
1,07828	0,52238	0,48252	0,23295	0,26888	0,41519	0,41027	1,99255	0,01326	1,16152
2,20542	2,024	0,62365	0,74987	1,7568	0,87015	1,26575	0,53014	0,68218	0,74924
0,7493	0,79962	1,25879	0,22186	2,24813	2,09381	0,88433	0,59806	0,34916	1,6006
0,37514	0,25735	0,36477	0,88158	0,51657	0,24283	0,81136	0,54823	0,05998	0,45036
2,39269	0,11661	1,05747	1,15535	0,49978	1,11133	0,95951	0,84071	1,22188	0,04744
0,22611	0,38573	0,70115	0,98244	1,60858	0,81654	0,07279	0,02385	0,50645	2,61476
0,9701	1,60596	0,70741	0,26846	0,22006	3,79664	0,28622	0,0279	0,45234	0,73708
0,38262	0,15533	0,38678	0,12341	0,19473	0,00985	0,83646	0,04656	0,22416	1,04613
0,47536	0,25044	0,06814	0,66754	0,47073	0,07505	0,14197	1,17069	1,02279	0,00012
0,55476	0,22297	0,46444	1,53611	0,94401	0,45698	1,60471	0,31486	0,7945	1,16895
1,33171	0,61808	1,12193	1,97371	0,95524	1,23751	0,34829	2,21917	0,06421	0,44362
0,4341	2,39495	0,34261	0,62867	0,12646	0,40653	0,59913	1,09162	0,76738	0,68827
1,33814	1,04889	0,50298	1,4459	0,83232	0,87615	0,3547	0,58477	1,68206	1,80308
0,78406	0,65326	0,26612	0,44836	0,7612	0,00208	0,93547	0,16637	0,99968	0,71624
0,22226	0,77007	0,09831	0,26292	0,1678	0,66263	0,04692	0,34376	0,84355	0,36237

Упорядоченная выборка

0,00012	0,00208	0,00985	0,01326	0,02226	0,02385	0,02491	0,0279	0,03376	0,04656
0,04692	0,04744	0,05998	0,06421	0,06814	0,07279	0,07505	0,08173	0,08407	0,09831
0,10275	0,10856	0,11661	0,11753	0,12341	0,12646	0,14197	0,15533	0,16637	0,1678
0,1714	0,19473	0,21905	0,22006	0,22186	0,22226	0,22297	0,22416	0,22611	0,23295
0,24283	0,25044	0,25626	0,25735	0,26292	0,26439	0,26612	0,26846	0,26888	0,27751
0,28622	0,31486	0,34261	0,34376	0,34829	0,34916	0,3547	0,36237	0,36477	0,37514
0,38153	0,38262	0,38573	0,38678	0,39415	0,40653	0,41027	0,41519	0,4341	0,44362
0,44836	0,45036	0,45234	0,45393	0,45698	0,46444	0,47073	0,47536	0,48252	0,48437
0,49385	0,49978	0,50298	0,50645	0,51657	0,52196	0,52238	0,52722	0,52902	0,53014
0,53269	0,54266	0,54823	0,55476	0,58477	0,59806	0,59913	0,61679	0,61808	0,62365
0,62867	0,63648	0,65326	0,66263	0,66328	0,66754	0,67258	0,68218	0,68827	0,70115
0,70238	0,70741	0,71624	0,73708	0,74924	0,7493	0,74987	0,75013	0,7612	0,76738
0,77007	0,78406	0,7945	0,79823	0,79962	0,81136	0,81654	0,83232	0,83646	0,83937
0,84071	0,84355	0,87015	0,87149	0,87615	0,88158	0,88433	0,89313	0,93547	0,94401
0,95524	0,95951	0,9701	0,98244	0,99968	1,01997	1,02279	1,0284	1,04613	1,04889
1,05011	1,05747	1,05899	1,07828	1,09162	1,09834	1,11133	1,12193	1,15535	1,16019
1,16152	1,16895	1,17069	1,20179	1,22188	1,23751	1,25879	1,26575	1,33171	1,33814
1,43742	1,4459	1,50939	1,53611	1,6006	1,60471	1,60596	1,60858	1,61255	1,68206
1,7568	1,80308	1,89257	1,97371	1,97856	1,99197	1,99255	2,00839	2,024	2,09381
2,20542	2,21917	2,24813	2,39269	2,39495	2,61476	3,2017	3,36295	3,62593	3,79664

Группированная выборка(интервальный вариационный ряд)

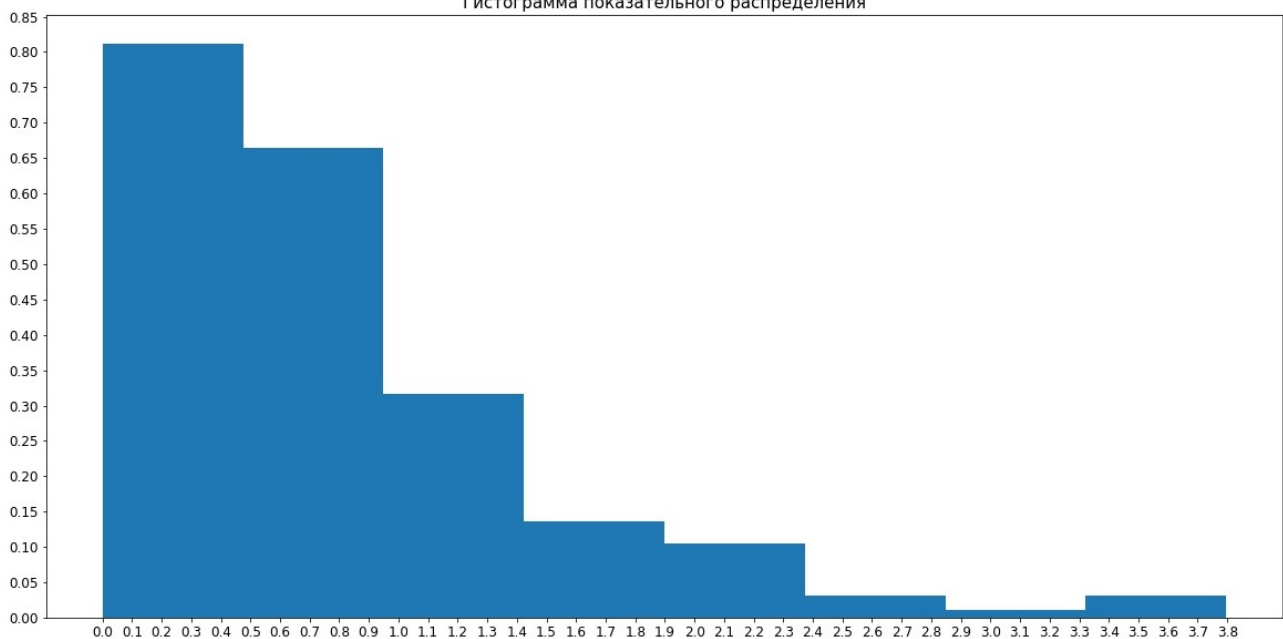
Интервалы	n_j	w_j
[0 , 0.47458]	77	0.385
(0.47458 , 0.94916]	63	0.315
(0.94916 , 1.42374]	30	0.15
(1.42374 , 1.89832]	13	0.065
(1.89832 , 2.3729]	10	0.05
(2.3729 , 2.84748]	3	0.015
(2.84748 , 3.32206]	1	0.005
(3.32206 , 3.79664]	3	0.015
	200	1

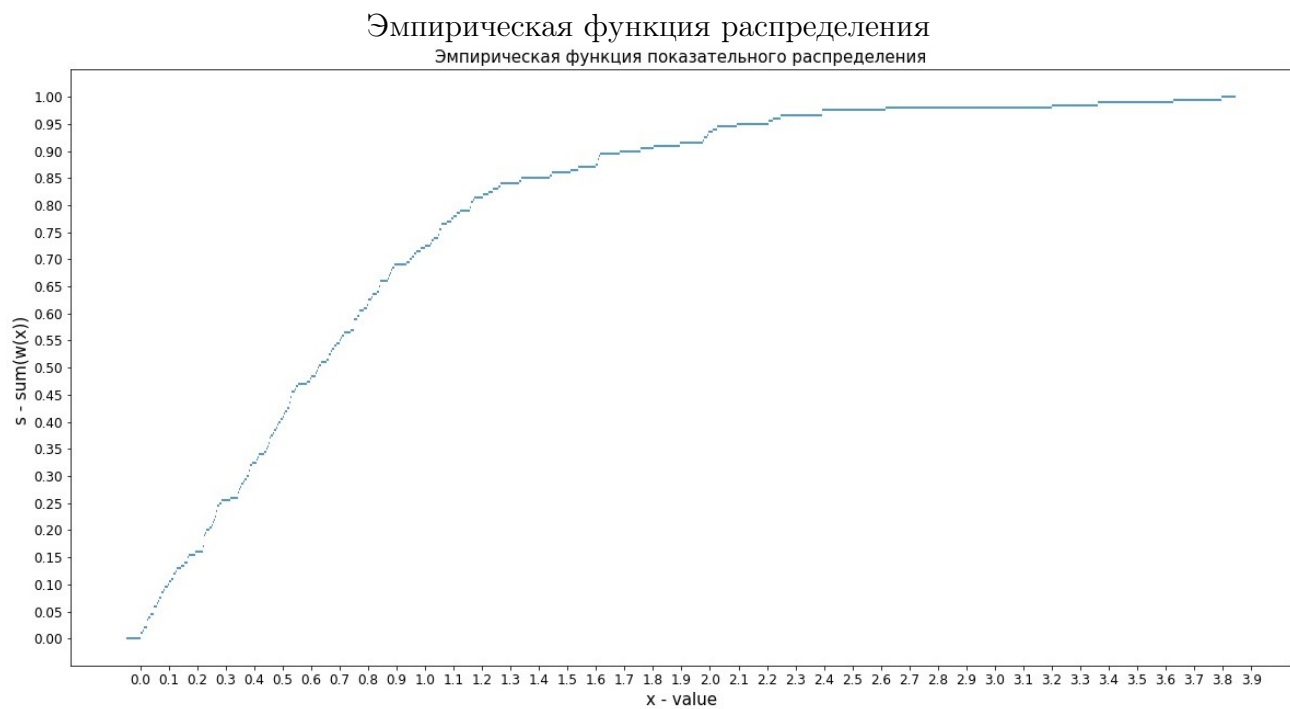
Ассоциированный статистический ряд

x^*_i	n^*_j	w^*_j
0.23729	77	0.385
0.71187	63	0.315
1.18645	30	0.15
1.66103	13	0.065
2.13561	10	0.05
2.61019	3	0.015
3.08477	1	0.005
3.55935	3	0.015
	200	1

Гистограмма относительных частот

Гистограмма показательного распределения





Результаты расчётов требуемых характеристик:

- Выборочное среднее: 0.81628;
- Выборочная дисперсия с поправкой Шеппарда: 0.45457;
- Выборочное среднее квадратическое отклонение: 0.67422;
- Выборочная мода: 0.40157;
- Выборочная медиана: 0.64784;
- Выборочный коэффициент асимметрии: 1.76555;
- Выборочный коэффициент эксцесса: 3.61917;

3.3 Задание №3

$$a = -3.35 \quad q = 2.65 \quad N = 200$$

Полученная выборка

1,89152	2,15713	-0,82834	-1,12209	1,13595	-3,25676	0,85771	-0,08543	-0,02199	2,12126
1,25428	-1,99151	-0,15651	-0,15417	-1,99588	2,49826	-2,93368	1,84007	0,43262	1,9926
-0,71219	1,94179	-0,05445	-2,785	-1,05679	1,69977	0,15106	1,75528	-2,6311	1,49349
1,46536	-0,08976	0,89638	-1,065	-0,73495	-0,50802	-1,3019	0,27751	0,38669	-1,78455
-1,09045	1,01867	-0,38311	1,7912	-1,93848	1,86916	-1,40854	2,35514	-2,2798	-0,85877
2,11927	0,5975	1,32619	-0,19375	0,16656	0,73287	1,29716	0,81028	0,83105	-2,91486
-1,36938	0,89113	-2,44871	-1,98074	0,32943	1,41372	-2,92609	-0,13699	2,49127	0,88636
-1,78839	1,17413	0,29088	1,87154	-2,31283	1,93481	1,1418	-2,69223	1,6476	1,1512
-3,07708	-3,12639	-1,18972	-1,71956	-1,30758	-2,67817	0,03499	0,82039	-2,09125	-2,38511
2,09813	-2,54193	-0,01218	0,80003	-2,65041	1,3563	-2,74763	1,59642	1,65053	-0,08036
1,50969	-3,1459	1,74116	-1,78309	-0,75873	-1,75045	-0,17487	1,58779	1,99416	-0,28403
-0,19025	-1,76712	-1,3234	-1,81649	-0,45266	1,61417	0,60884	0,97134	-0,76623	-2,76586
-2,19765	-0,18768	-1,66378	2,2599	1,42717	1,69739	2,32577	-2,50846	-2,5373	-2,30608
-0,53511	-2,78798	-1,03106	-0,38332	-0,10884	0,64584	0,96004	0,51477	1,14641	-1,01092
1,02632	-3,0366	2,38775	-2,9251	-1,71487	0,98852	-1,20019	-1,4212	-1,32185	2,37744
-0,05645	-0,36365	-0,60858	-2,46386	0,30849	-1,52628	0,80294	0,99123	-3,18676	-0,70646
2,28689	0,8463	1,52896	0,23351	2,261	1,43486	0,43302	-1,7528	-0,85751	-1,3299
-0,66516	0,88737	0,73674	1,63298	-2,55162	1,99287	-3,33647	-0,30062	2,20879	1,62867
-2,16086	-2,90983	-2,47474	-2,41753	-1,00481	-0,22539	-0,3293	1,96523	2,47544	-1,79682
0,36051	-0,21533	-1,8115	0,52693	-0,10376	1,30697	-0,18985	0,66531	-1,5711	-0,45399

Упорядоченная выборка

-3,33647	-3,25676	-3,18676	-3,1459	-3,12639	-3,07708	-3,0366	-2,93368	-2,92609	-2,9251
-2,91486	-2,90983	-2,78798	-2,785	-2,76586	-2,74763	-2,69223	-2,67817	-2,65041	-2,6311
-2,55162	-2,54193	-2,5373	-2,50846	-2,47474	-2,46386	-2,44871	-2,41753	-2,38511	-2,31283
-2,30608	-2,2798	-2,19765	-2,16086	-2,09125	-1,99588	-1,99151	-1,98074	-1,93848	-1,81649
-1,8115	-1,79682	-1,78839	-1,78455	-1,78309	-1,76712	-1,7528	-1,75045	-1,71956	-1,71487
-1,66378	-1,5711	-1,52628	-1,4212	-1,40854	-1,36938	-1,3299	-1,3234	-1,32185	-1,30758
-1,3019	-1,20019	-1,18972	-1,12209	-1,09045	-1,065	-1,05679	-1,03106	-1,01092	-1,00481
-0,85877	-0,85751	-0,82834	-0,76623	-0,75873	-0,73495	-0,71219	-0,70646	-0,66516	-0,60858
-0,53511	-0,50802	-0,45399	-0,45266	-0,38332	-0,38311	-0,36365	-0,3293	-0,30062	-0,28403
-0,22539	-0,21533	-0,19375	-0,19025	-0,18985	-0,18768	-0,17487	-0,15651	-0,15417	-0,13699
-0,10884	-0,10376	-0,08976	-0,08543	-0,08036	-0,05645	-0,05445	-0,02199	-0,01218	0,03499
0,15106	0,16656	0,23351	0,27751	0,29088	0,30849	0,32943	0,36051	0,38669	0,43262
0,43302	0,51477	0,52693	0,5975	0,60884	0,64584	0,66531	0,73287	0,73674	0,80003
0,80294	0,81028	0,82039	0,83105	0,8463	0,85771	0,88636	0,88737	0,89113	0,89638
0,96004	0,97134	0,98852	0,99123	1,01867	1,02632	1,13595	1,1418	1,14641	1,1512
1,17413	1,25428	1,29716	1,30697	1,32619	1,3563	1,41372	1,42717	1,43486	1,46536
1,49349	1,50969	1,52896	1,58779	1,59642	1,61417	1,62867	1,63298	1,6476	1,65053
1,69739	1,69977	1,74116	1,75528	1,7912	1,84007	1,86916	1,87154	1,89152	1,93481
1,94179	1,96523	1,9926	1,99287	1,99416	2,09813	2,11927	2,12126	2,15713	2,20879
2,2599	2,261	2,28689	2,32577	2,35514	2,37744	2,38775	2,47544	2,49127	2,49826

Группированная выборка(интервальный вариационный ряд)

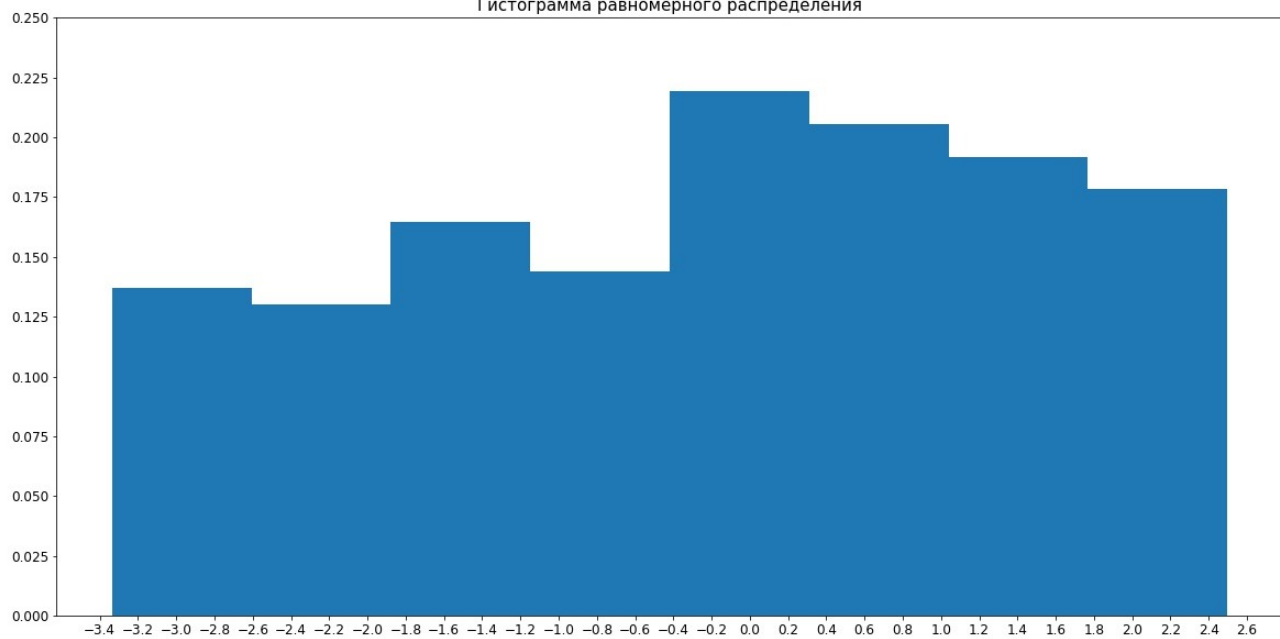
Интервалы	n_j	w_j
$[-3.35, -2.6]$	20	0.1
$(-2.6, -1.85]$	19	0.095
$(-1.85, -1.1]$	25	0.125
$(-1.1, -0.35]$	23	0.115
$(-0.35, 0.4]$	32	0.16
$(0.4, 1.15]$	30	0.15
$(1.15, 1.9]$	30	0.15
$(1.9, 2.65]$	21	0.105
	200	1

Ассоциированный статистический ряд

x^*_j	n'_j	w'_j
-2.975	20	0.1
-2.225	19	0.095
-1.475	25	0.125
-0.725	23	0.115
0.025	32	0.16
0.775	30	0.15
1.525	30	0.15
2.275	21	0.105
	200	1

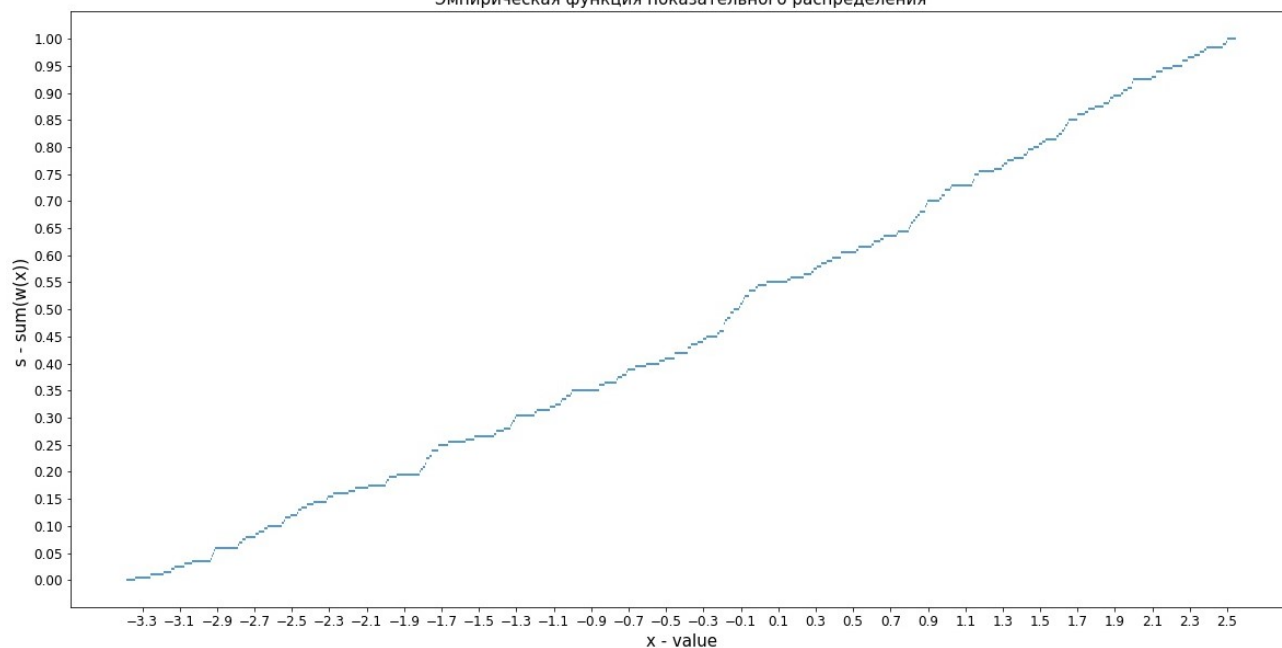
Гистограмма относительных частот

Гистограмма равномерного распределения



Эмпирическая функция распределения

Эмпирическая функция показательного распределения



Результаты расчётов требуемых характеристик:

- Выборочное среднее: -0.18875;
- Выборочная дисперсия с поправкой Шешпарда: 2.58775;
- Выборочное среднее квадратическое отклонение: 1.60865;
- Выборочная мода: 0.26364;
- Выборочная медиана: -0.04531;
- Выборочный коэффициент асимметрии: -0.18923;
- Выборочный коэффициент эксцесса: -1.01328;

4 Анализ результатов и выводы

Таблица сравнения относительных частот и теоретических вероятностей

Показательное распределение

Интервалы	w'_j	p_j	$ w'_j - p_j $
$[-5.42243, -4.2125]$	0.03	0.01473	0.01527
$(-4.2125, -3.00256]$	0.06	0.0643	0.0043
$(-3.00256, -1.79262]$	0.15	0.16947	0.01947
$(-1.79262, -0.58269]$	0.255	0.27013	0.01513
$(-0.58269, 0.62725]$	0.23	0.26051	0.03051
$(0.62725, 1.83718]$	0.21	0.152	0.058
$(1.83718, 3.04712]$	0.045	0.05362	0.00862
$(3.04712, 4.25706]$	0.02	0.01143	0.00857
	1	1	0.058

Нормальное распределение

Интервалы	w'_j	p_j	$ w'_j - p_j $
$[0, 0.47458]$	0.385	0.46804	0.08304
$(0.47458, 0.94916]$	0.315	0.24898	0.06602
$(0.94916, 1.42374]$	0.15	0.13245	0.01755
$(1.42374, 1.89832]$	0.065	0.07046	0.00546
$(1.89832, 2.3729]$	0.05	0.03748	0.01252
$(2.3729, 2.84748]$	0.015	0.01994	0.00494
$(2.84748, 3.32206]$	0.005	0.01061	0.00561
$(3.32206, 3.79664]$	0.015	0.00564	0.00936
	1	1	0.08304

Равномерное распределение

Интервалы	w'_j	p_j	$ w'_j - p_j $
$[-3.35, -2.6]$	0.1	0.125	0.025
$(-2.6, -1.85]$	0.095	0.125	0.03
$(-1.85, -1.1]$	0.125	0.125	0
$(-1.1, -0.35]$	0.115	0.125	0.01
$(-0.35, 0.4]$	0.16	0.125	0.035
$(0.4, 1.15]$	0.15	0.125	0.025
$(1.15, 1.9]$	0.15	0.125	0.025
$(1.9, 2.65]$	0.105	0.125	0.02
	1	1	0.035

Таблица сравнения рассчитанных характеристик с теоретическими значениями

Показательное распределение

Название показателя	Экспериментальное значение	Теоретическое значение	Абсолютное отклонение	Относительное отклонение
Выборочное среднее	-0.57664	-0.67	0.09336	0.13934
Выборочная дисперсия	3.09865	2.7889	0.30975	0.11107
Выборочное среднее квадратичное отклонение	1.7603	1.67	0.0903	0.05407
Выборочная мода	-0.81537	-0.67	0.14537	0.21697
Выборочная медиана	-0.55639	-0.67	0.11361	0.16957
Выборочный коэффициент асимметрии	-0.16131	0	0.16131	-
Выборочный коэффициент эксцесса	0.00593	0	0.00593	-

Нормальное распределение

Название показателя	Экспериментальное значение	Теоретическое значение	Абсолютное отклонение	Относительное отклонение
Выборочное среднее	0.81628	0.75188	0.0644	0.08565
Выборочная дисперсия	0.45457	0.56532	0.11076	0.19592
Выборочное среднее квадратичное отклонение	0.67422	0.75188	0.07766	0.10329
Выборочная мода	0.40157	0	0.40157	-
Выборочная медиана	0.64784	0.52116	0.12668	0.24307
Выборочный коэффициент асимметрии	1.76555	2	0.23445	0.11722
Выборочный коэффициент эксцесса	3.61917	6	2.38083	0.39681

Равномерное распределение

Название показателя	Экспериментальное значение	Теоретическое значение	Абсолютное отклонение	Относительное отклонение
Выборочное среднее	-0.18875	-0.35	0.16125	0.46071
Выборочная дисперсия	2.58775	3	0.41225	0.13742
Выборочное среднее квадратичное отклонение	1.60865	1.73205	0.1234	0.07125
Выборочная мода	0.26364	-0.35	0.61364	1.75326
Выборочная медиана	-0.04531	-0.35	0.30469	0.87054
Выборочный коэффициент асимметрии	-0.18923	0	0.18923	-
Выборочный коэффициент эксцесса	-1.01328	-1.2	0.18672	0.1556

5 Список использованной 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.
8. Кобзарь А.И. Прикладная математическая статистика: Для инженеров и научных работников — М.: ФИЗМАТЛИТ, 2006. — 816 с.
9. Монсик В.Б., Скрынников А. А. Вероятность и статистика.— М. : БИНОМ, 2015 — 384 с.
10. Сборник задач по теории вероятностей, математической статистике и теории случайных функций: Учеб. пособие для вузов / Под ред. А. А. Свешникова. — СПб.: Лань, 2012. — 472 с.
11. Письменный Д.Т. Конспект лекций по теории вероятностей, математической статистике и случайным процессам: учеб. пособие для вузов. — М.: Айрис-пресс, 2013. — 288 с.
12. Ramachandran Kandethody M., Tsokos Chris P. Mathematical Statistics with Applications in R. — N-Y.: Academic Press, 2009. — 826 p.
13. Ивановский Р.И. Теория вероятностей и математическая статистика: Основы, прикладные аспекты с примерами и задачами в среде Mathcad:Учеб. пособие для вузов — СПб.: БХВ-Петербург, 2008. — 528 с.

```

import matplotlib.pyplot as plt
import numpy as np
import math
import plotly.graph_objects as go
import copy
import xlswriter

N=200
v=67
mu=(-1)**v*0.01*v #A
L=2+v*(0.01)*((-1)**v) #lambda
s_d=1+(0.01*v) #standard deviation
a=(-1)**v*0.05*v
b=a+6

#beta=1/L
#Normal=np.random.normal(mu, s_d, N)
#Indicative=np.random.exponential(beta, N)
#uniform=np.random.uniform(a,b, N)

#np.savetxt('normal.txt', Normal,'%f')
#np.savetxt('indicative.txt', Indicative,'%f')
#np.savetxt('uniform.txt', uniform,'%f')

my_file = open("normal.txt", "r")
Normal=[]
for num in my_file:
    Normal.append(float(num))
my_file.close()
Normal_unsort = []
Normal_unsort = copy.deepcopy(Normal)
my_file1 = open("indicative.txt", "r")
Indicative=[]
for num in my_file1:
    Indicative.append(float(num))
my_file1.close()
Indicative_unsort = []
Indicative_unsort = copy.deepcopy(Indicative)
my_file2 = open("uniform.txt", "r")
uniform=[]
for num in my_file2:
    uniform.append(float(num))
my_file2.close()
uniform_unsort = []
uniform_unsort = copy.deepcopy(uniform)

Normal.sort()
Indicative.sort()
uniform.sort()

m=1+int(math.log(N,2))
n_0=min(Normal)
n_m=max(Normal)
d_n=n_m-n_0
i_0=0
i_m=max(Indicative)
d_i=i_m-i_0

```

```

d_u=b-a

a_n=[n_0]
for i in range(m-1):
    a_n.append(a_n[i]+d_n/m)
a_n.append(n_m)

a_i=[i_0]
for i in range(m-1):
    a_i.append(a_i[i]+d_i/m)
a_i.append(i_m)

a_u=[a]
for i in range(m-1):
    a_u.append(a_u[i]+d_u/m)
a_u.append(b)

n_n=[]
counter = 0
k=1
helper = a_n[k]
for i in range(len(Normal)):
    if Normal[i]<=helper:
        counter += 1
    else:
        n_n.append(counter)
        k += 1
        helper = a_n[k]
        if (Normal[i]<=helper):
            counter=1
        else: counter=0
n_n.append(counter)

n_i=[]
counter = 0
k=1
helper = a_i[k]
for i in range(len(Indicative)):
    if Indicative[i]<=helper:
        counter += 1
    else:
        n_i.append(counter)
        k += 1
        helper = a_i[k]
        if (Indicative[i]<=helper):
            counter=1
        else:
            counter=0
n_i.append(counter)

n_u=[]
counter = 0
k=1
helper = a_u[k]
for i in range(len(uniform)):
    if uniform[i]<=helper:
        counter += 1

```



```

else:
    n_u.append(counter)
    k += 1
    helper = a_u[k]
    if (uniform[i]<=helper):
        counter=1
    else: counter=0
n_u.append(counter)
w_n=[]
for i in range(len(n_n)):
    w_n.append(n_n[i]/N)

w_i=[]
for i in range(len(n_i)):
    w_i.append(n_i[i]/N)

w_u=[]
for i in range(len(n_u)):
    w_u.append(n_u[i]/N)

x_n = []
for i in range(len(a_n)-1):
    x_n.append((a_n[i]+a_n[i+1])/2)

x_i = []
for i in range(len(a_i)-1):
    x_i.append((a_i[i]+a_i[i+1])/2)

x_u = []
for i in range(len(a_u)-1):
    x_u.append((a_u[i]+a_u[i+1])/2)

math_exp_n=0
for i in range(len(x_n)):
    math_exp_n+=(x_n[i]*w_n[i])
print(math_exp_n)

math_exp_i=0
for i in range(len(x_i)):
    math_exp_i+=(x_i[i]*w_i[i])
print(math_exp_i)

math_exp_u=0
for i in range(len(x_u)):
    math_exp_u+=(x_u[i]*w_u[i])
print(math_exp_u)

h_n=(n_m-n_0)/m
h_i=(i_m-i_0)/m
h_u=(b-a)/m

Disp_n=0
for i in range(len(x_n)):
    Disp_n+=w_n[i]*((x_n[i]-math_exp_n)**2)
Disp_n=(h_n**2)/12
print(Disp_n)

```

```

Disp_i=0
for i in range(len(x_i)):
    Disp_i+=w_i[i]*((x_i[i]-math_exp_i)**2)
Disp_i=(h_i**2)/12
print(Disp_i)

Disp_u=0
for i in range(len(x_u)):
    Disp_u+=w_u[i]*((x_u[i]-math_exp_u)**2)
Disp_u=(h_u**2)/12
print(Disp_u)

stand_dev_n=math.sqrt(Disp_n)
stand_dev_i=math.sqrt(Disp_i)
stand_dev_u=math.sqrt(Disp_u)

k=n_n.index(max(n_n))
Moda_n=a_n[k]+(h_n*(w_n[k]-w_n[k-1]))/(2*w_n[k]-w_n[k-1]-w_n[k+1]))
s_n=0
i=0
Med_n=0
while s_n<0.5:
    s_n+=w_n[i]
    i+=1
if s_n==0.5:
    Med_n=a_n[i]
else:
    s_n=w_n[i-1]
    Med_n=a_n[i-1]+(h_n/w_n[i-1])*(0.5-s_n)
print(Moda_n)
print(Med_n,'\n')

k=n_i.index(max(n_i))
Moda_i=a_i[k]+(h_i*(w_i[k]-w_i[k-1]))/(2*w_i[k]-w_i[k-1]-w_i[k+1]))
s_i=0
i=0
Med_i=0
while s_i<0.5:
    s_i+=w_i[i]
    i+=1
if s_i==0.5:
    Med_i=a_i[i]
else:
    s_i=w_i[i-1]
    Med_i=a_i[i-1]+(h_i/w_i[i-1])*(0.5-s_i)
print(Moda_i)
print(Med_i,'\n')

k=n_u.index(max(n_u))
Moda_u=a_u[k]+(h_u*(w_u[k]-w_u[k-1]))/(2*w_u[k]-w_u[k-1]-w_u[k+1]))
s_u=0
i=0
Med_u=0
while s_u<0.5:
    s_u+=w_u[i]
    i+=1
if s_u==0.5:

```

```

    Med_u=a_u[i]
else:
    s_u=w_u[i-1]
    Med_u=a_u[i-1]+(h_u/w_u[i-1])*(0.5-s_u)
print(Moda_u)
print(Med_u,'\n')
c_m_3_n=0
c_m_4_n=0
for i in range(len(x_n)):
    c_m_3_n+=w_n[i]*((x_n[i]-math_exp_n)**3)
    c_m_4_n+=w_n[i]*((x_n[i]-math_exp_n)**4)
c_as_n=c_m_3_n/(stand_dev_n**3)
c_ex_n= (c_m_4_n/(stand_dev_n**4))-3)

c_m_3_i=0
c_m_4_i=0

for i in range(len(x_i)):
    c_m_3_i+=w_i[i]*((x_i[i]-math_exp_i)**3)
    c_m_4_i+=w_i[i]*((x_i[i]-math_exp_i)**4)

c_as_i=c_m_3_i/(stand_dev_i**3)
c_ex_i= ((c_m_4_i/(stand_dev_i**4))-3)

c_m_3_u=0
c_m_4_u=0

for i in range(len(x_u)):
    c_m_3_u+=w_u[i]*((x_u[i]-math_exp_u)**3)
    c_m_4_u+=w_u[i]*((x_u[i]-math_exp_u)**4)

c_as_u=c_m_3_u/(stand_dev_u**3)
c_ex_u= ((c_m_4_u/(stand_dev_u**4))-3)

S=[1/N]
for i in range(1,N):
    S.append(S[i-1]+(1/N))

fig = plt.figure(figsize=(20, 10))
plt.hlines(0, Normal[0]-0.05, Normal[0])
for i in range(len(S)-1):
    plt.hlines(S[i], Normal[i], Normal[i+1])
plt.hlines(S[i+1], Normal[i+1], Normal[i+1]+0.05)
plt.title("Эмпирическая функция нормального распределения",fontsize =15)
plt.ylabel('s - sum(w(x)) ', fontsize =15)
plt.yticks(np.arange(0.0, 1.05, step=0.05), fontsize =12)
plt.xticks(np.arange(round(Normal[0],1), round(Normal[len(Normal)-1]+0.3,1), step=0.3), fontsize =12)
plt.xlabel('x - value', fontsize =15)
plt.savefig('normal_emp.jpg')

fig = plt.figure(figsize=(20, 10))
plt.hlines(0, -0.05, Indicative[0])
for i in range(len(S)-1):
    plt.hlines(S[i], Indicative[i], Indicative[i+1])
plt.hlines(S[i+1], Indicative[i+1], Indicative[i+1]+0.05)
plt.title("Эмпирическая функция показательного распределения",fontsize =15)
plt.ylabel('s - sum(w(x)) ', fontsize =15)

```

```

plt.yticks(np.arange(0.0, 1.05, step=0.05), fontsize =12)
plt.xticks(np.arange(round(Indicative[0],1), round(Indicative[len(Indicative)-1],1)+0.2, step=0.1),
fontsize =12)
plt.xlabel('x - value', fontsize =15)
plt.savefig('indicative_emp.jpg')

fig = plt.figure(figsize=(20, 10))
plt.hlines(0, uniform[0]-0.05, uniform[0])
for i in range(len(S)-1):
    plt.hlines(S[i], uniform[i], uniform[i+1])
plt.hlines(S[i+1], uniform[i+1], uniform[i+1]+0.05)
plt.title('Эмпирическая функция показательного распределения',fontsize =15)
plt.ylabel('s - sum(w(x)) ', fontsize =15)
plt.yticks(np.arange(0.0, 1.05, step=0.05), fontsize =12)
plt.xticks(np.arange(round(uniform[0],1), round(uniform[len(uniform)-1]+0.2,1), step=0.2), fontsize
=12)
plt.xlabel('x - value', fontsize =15)
plt.savefig('uniform_emp.jpg')

y_n=[]
for i in range(len(w_n)):
    y_n.append(w_n[i]/h_n)

fig = plt.figure(figsize=(25, 15))
plt.hist(Normal,bins=m, density=True)
plt.title('Гистограмма нормального распределения',fontsize =15)
plt.yticks(np.arange(0.0, 0.250, step=0.025), fontsize =12)
plt.xticks(np.arange(round(Normal[0]-0.1,1), round(Normal[len(Normal)-1]+0.1,1), step=0.2), fontsize
=10)
plt.savefig('normal_hist.jpg')

fig = plt.figure(figsize=(20, 10))
plt.hist(Indicative,bins=m, density=True)
plt.title('Гистограмма показательного распределения',fontsize =15)
plt.yticks(np.arange(0.0, 0.9, step=0.05), fontsize =12)
plt.xticks(np.arange(round(Indicative[0],1), round(Indicative[len(Indicative)-1]+0.1,1), step=0.1),
fontsize =12)
plt.savefig('indicative_hist.jpg')

fig = plt.figure(figsize=(20, 10))
plt.hist(uniform,bins=m, density=True)
plt.title('Гистограмма равномерного распределения',fontsize =15)
plt.yticks(np.arange(0.0, 0.275, step=0.025), fontsize =12)
plt.xticks(np.arange(round(uniform[0],1)-0.1, round(uniform[len(uniform)-1]+0.2,1), step=0.2), fontsize
=12)
plt.savefig('uniform_hist.jpg')

def normal_prob(miu, stdev, x):
    return 0.5 * (1 + math.erf((x-miu)/(stdev * 2**0.5)))

P_n=[]
for i in range(len(a_n)-1):
    P_n.append(normal_prob(mu,s_d,a_n[i+1])-normal_prob(mu,s_d,a_n[i]))

P_i=[]
for i in range(len(a_i)-1):
    P_i.append(math.exp(-a_i[i]*L)-math.exp(-a_i[i+1]))

```

```

P_u=[]
for i in range(len(a_u)-1):
    P_u.append((a_u[i+1]-a_u[i])/(b-a))

Disp_n_t=s_d**2
c_as_i_t=2
c_ex_i_t=6
Med_i_t=math.log(2)/L
s_d_i_t=L**(-1)
Disp_i_t=L**(-2)
math_exp_i_t=L**(-1)
Moda_i_t=0

c_as_u_t=0
c_ex_u_t=-6/5
Moda_u_t=(a+b)/2
Med_u_t=(a+b)/2
s_d_u_t=(b-a)/(2*math.sqrt(3))
Disp_u_t=((b-a)**2)/12
math_exp_u_t=(a+b)/2

a_n_h=[]
w_n_h=[]
P_n_h=[]
abs_n_w_p_h=[]
h="["+str(round(a_n[0],5))+", "+str(round(a_n[1],5))+"]"
a_n_h.append(h)
for i in range(1,len(a_n)-1):
    a_n_h.append("["+str(round(a_n[i],5))+", "+str(round(a_n[i+1],5))+"]")
a_n_h.append(' ')
for i in range(len(w_n)):
    w_n_h.append(round(w_n[i], 5))
w_n_h.append(sum(w_n))
for i in range(len(P_n)):
    P_n_h.append(round(P_n[i], 5))
P_n_h.append(round(sum(P_n)))
for i in range(len(w_n)):
    abs_n_w_p_h.append(round((abs(w_n[i]-P_n[i])), 5))
max_n_delt = max(abs_n_w_p_h)
abs_n_w_p_h.append(max_n_delt)

x_n_h=[]
n_n_h=[]
for i in range(len(x_n)):
    x_n_h.append(round(x_n[i],5))
x_n_h.append(' ')
for i in range(len(n_n)):
    n_n_h.append(round(n_n[i], 5))
n_n_h.append(sum(n_n))

fig = go.Figure(data=[go.Table(header=dict(values=["Интервалы", 'n_j', 'w_j']),
cells=dict(values=[a_n_h, n_n_h, w_n_h]))
])
fig.show()

fig = go.Figure(data=[go.Table(header=dict(values=["Интервалы", 'w_j', 'p_j', '|w_j - p_j|']),
cells=dict(values=[a_n_h, w_n_h, P_n_h, abs_n_w_p_h]))
])

```

```

])
fig.show()

a_i_h=[]
w_i_h=[]
P_i_h=[]
abs_i_w_p_h=[]
a_i_h.append("[ "+str(round(a_i[0],5))+", "+str(round(a_i[1],5))+"]")
for i in range(1,len(a_i)-1):
    a_i_h.append("(" +str(round(a_i[i],5))+", "+str(round(a_i[i+1],5))+")")
a_i_h.append(' ')
for i in range(len(w_i)):
    w_i_h.append(round(w_i[i], 5))
w_i_h.append(sum(w_i))
for i in range(len(P_i)):
    P_i_h.append(round(P_i[i], 5))
P_i_h.append(round(sum(P_i)))
for i in range(len(w_i)):
    abs_i_w_p_h.append(round((abs(w_i[i]-P_i[i])), 5))
max_i_delt = max(abs_i_w_p_h)
abs_i_w_p_h.append(max_i_delt)

x_i_h=[]
n_i_h=[]
for i in range(len(x_i)):
    x_i_h.append(round(x_i[i],5))
x_i_h.append(' ')
for i in range(len(n_i)):
    n_i_h.append(round(n_i[i], 5))
n_i_h.append(sum(n_i))

fig = go.Figure(data=[go.Table(header=dict(values=["Интервалы", 'n_j', 'w_j']),
cells=dict(values=[a_i_h, n_i_h, w_i_h]))
])
fig.show()

fig = go.Figure(data=[go.Table(header=dict(values=["Интервалы", 'w_j', 'p_j', '|w_j - p_j|']),
cells=dict(values=[a_i_h, w_i_h, P_i_h, abs_i_w_p_h]))
])
fig.show()

a_u_h=[]
w_u_h=[]
P_u_h=[]
abs_u_w_p_h=[]
a_u_h.append("[ "+str(round(a_u[0],5))+", "+str(round(a_u[1],5))+"]")
for i in range(1,len(a_u)-1):
    a_u_h.append("(" +str(round(a_u[i],5))+", "+str(round(a_u[i+1],5))+")")
a_u_h.append(' ')
for i in range(len(w_u)):
    w_u_h.append(round(w_u[i], 5))
w_u_h.append(sum(w_u))
for u in range(len(P_u)):
    P_u_h.append(round(P_u[i], 5))
P_u_h.append(round(sum(P_u)))
for i in range(len(w_u)):
    abs_u_w_p_h.append(round((abs(w_u[i]-P_u[i])), 5))

```

```

abs_u_w_p_h.append(max(abs_u_w_p_h))

x_u_h=[]
n_u_h=[]
for i in range(len(x_u)):
    x_u_h.append(round(x_u[i],5))
x_u_h.append(' ')
for i in range(len(n_u)):
    n_u_h.append(round(n_u[i], 5))
n_u_h.append(sum(n_u))

fig = go.Figure(data=[go.Table(header=dict(values=["Интервалы", 'n`_j' , 'w`_j']),
cells=dict(values=[a_u_h, n_u_h, w_u_h]))
])
fig.show()

fig = go.Figure(data=[go.Table(header=dict(values=["Интервалы", 'w`_j' , 'p`_j', '|w`_j - p`_j|']),
cells=dict(values=[a_u_h, w_u_h, P_u_h, abs_u_w_p_h]))
])
fig.show()

fig = go.Figure(data=[go.Table(header=dict(values=["x*_i", 'n`_j', 'w`_j']),
cells=dict(values=[x_n_h, n_n_h, w_n_h]))
])
fig.show()

fig = go.Figure(data=[go.Table(header=dict(values=["x*_i", 'n`_j', 'w`_j']),
cells=dict(values=[x_i_h, n_i_h, w_i_h]))
])
fig.show()

names = ['Выборочное среднее', 'Выборочная дисперсия', 'Выборочное среднее квадратичное
отклонение', 'Выборочная мода', 'Выборочная медиана', 'Выборочный коэффициент асимметрии',
'Выборочный коэффициент эксцесса']

experimental_n = [round(math_exp_n, 5), round(Disp_n, 5), round(stand_dev_n, 5), round(Moda_n, 5),
round(Med_n, 5), round(c_as_n, 5), round(c_ex_n, 5)]
theoretical_n = [round(mu, 5), round(Disp_n_t, 5), round(s_d, 5), mu, mu, 0, 0]
absolute_deviation_n = [round(abs(math_exp_n-mu), 5), round(abs(Disp_n - Disp_n_t), 5),
round(abs(stand_dev_n-s_d), 5), round(abs(Moda_n-mu), 5), round(abs(Med_n-mu), 5),
round(abs(c_as_n-0), 5), round(abs(c_ex_n-0), 5)]
relative_deviation_n = []
for i in range(len(absolute_deviation_n)):
    if(theoretical_n[i] == 0):
        relative_deviation_n.append(' - ')
    else:
        relative_deviation_n.append(round(abs(absolute_deviation_n[i]/theoretical_n[i]) , 5))

fig = go.Figure(data=[go.Table(header=dict(values=['Название показателя', 'Экспериментальное
значение', 'Теоретическое значение', 'Абсолютное отклонение', 'Относительное отклонение']),
cells=dict(values=[names, experimental_n, theoretical_n, absolute_deviation_n, relative_deviation_n])
])
fig.show()

experimental_i = [round(math_exp_i, 5), round(Disp_i, 5), round(stand_dev_i, 5), round(Moda_i, 5),
round(Med_i, 5), round(c_as_i, 5), round(c_ex_i, 5)]

```

```

theoretical_i = [round(math_exp_i_t, 5), round(Disp_i_t, 5), round(s_d_i_t, 5), Moda_i_t,
round(Med_i_t, 5), c_as_i_t, c_ex_i_t]
absolute_deviation_i = [round(abs(math_exp_i-math_exp_i_t), 5), round(abs(Disp_i - Disp_i_t), 5),
round(abs(stand_dev_i-s_d_i_t), 5), round(abs(Moda_i-Moda_i_t), 5), round(abs(Med_i-Med_i_t), 5),
round(abs(c_as_i-c_as_i_t), 5), round(abs(c_ex_i-c_ex_i_t), 5)]
relative_deviation_i = []
for i in range(len(absolute_deviation_i)):
    if(theoretical_i[i] == 0):
        relative_deviation_i.append(' - ')
    else:
        relative_deviation_i.append(round(abs(absolute_deviation_i[i]/theoretical_i[i]) , 5))

```

```

fig = go.Figure(data=[go.Table(header=dict(values=['Название показателя', 'Экспериментальное
значение', 'Теоретическое значение', 'Абсолютное отклонение', 'Относительное отклонение']),
cells=dict(values=[names, experimental_i, theoretical_i, absolute_deviation_i, relative_deviation_i]))
])
fig.show()

```

```

experimental_u = [round(math_exp_u, 5), round(Disp_u, 5), round(stand_dev_u, 5), round(Moda_u, 5),
round(Med_u, 5), round(c_as_u, 5), round(c_ex_u, 5)]
theoretical_u = [round(math_exp_u_t, 5), round(Disp_u_t, 5), round(s_d_u_t, 5), round(Moda_u_t, 5),
round(Med_u_t, 5), c_as_u_t, c_ex_u_t]
absolute_deviation_u = [round(abs(math_exp_u-math_exp_u_t), 5), round(abs(Disp_u - Disp_u_t), 5),
round(abs(stand_dev_u-s_d_u_t), 5), round(abs(Moda_u-Moda_u_t), 5), round(abs(Med_u-Med_u_t),
5), round(abs(c_as_u-c_as_u_t), 5), round(abs(c_ex_u-c_ex_u_t), 5)]
relative_deviation_u = []
for i in range(len(absolute_deviation_u)):
    if(theoretical_u[i] == 0):
        relative_deviation_u.append(' - ')
    else:
        relative_deviation_u.append(round(abs(absolute_deviation_u[i]/theoretical_u[i]) , 5))

```

```

fig = go.Figure(data=[go.Table(header=dict(values=['Название показателя', 'Экспериментальное
значение', 'Теоретическое значение', 'Абсолютное отклонение', 'Относительное отклонение']),
cells=dict(values=[names, experimental_u, theoretical_u, absolute_deviation_u, relative_deviation_u]))
])
fig.show()

```

```

matrix_v_n_s=[[0] * 20 for i in range(10)]
matrix_v_n_uns=[[0] * 20 for i in range(10)]
matrix_v_i_s=[[0] * 20 for i in range(10)]
matrix_v_i_uns=[[0] * 20 for i in range(10)]
matrix_v_u_s=[[0] * 20 for i in range(10)]
matrix_v_u_uns=[[0] * 20 for i in range(10)]
for i in range(10):
    for j in range(20):
        matrix_v_n_s[i][j]=round(Normal[i+j*10],5)
        matrix_v_n_uns[i][j]=round(Normal_unsort[i+j*10],5)
        matrix_v_i_s[i][j]=round(Indicative[i+j*10],5)
        matrix_v_i_uns[i][j]=round(Indicative_unsort[i+j*10],5)
        matrix_v_u_s[i][j]=round(uniform[i+j*10],5)
        matrix_v_u_uns[i][j]=round(uniform_unsort[i+j*10],5)

```

```

workbook = xlswriter.Workbook('Normal.xlsx')
worksheet = workbook.add_worksheet()

```

```

row = 0

```



```
for col, data in enumerate(matrix_v_n_s):
    worksheet.write_column(row, col, data)

workbook.close()

workbook = xlswriter.Workbook('Normal_unsort.xlsx')
worksheet = workbook.add_worksheet()

row = 0

for col, data in enumerate(matrix_v_n_uns):
    worksheet.write_column(row, col, data)

workbook.close()

workbook = xlswriter.Workbook('Indicative.xlsx')
worksheet = workbook.add_worksheet()

row = 0

for col, data in enumerate(matrix_v_i_s):
    worksheet.write_column(row, col, data)

workbook.close()

workbook = xlswriter.Workbook('Indicative_unsort.xlsx')
worksheet = workbook.add_worksheet()

row = 0

for col, data in enumerate(matrix_v_i_uns):
    worksheet.write_column(row, col, data)

workbook.close()

workbook = xlswriter.Workbook('uniform.xlsx')
worksheet = workbook.add_worksheet()

row = 0

for col, data in enumerate(matrix_v_u_s):
    worksheet.write_column(row, col, data)

workbook.close()

workbook = xlswriter.Workbook('uniform_unsort.xlsx')
worksheet = workbook.add_worksheet()

row = 0

for col, data in enumerate(matrix_v_u_uns):
    worksheet.write_column(row, col, data)

workbook.close()
```