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

ИНСТИТУТ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

КАФЕДРА ВЫСШЕЙ МАТЕМАТИКИ

Типовой расчет 2

по курсу **«Специальные методы моделирования»**

Тема: **Моделирование непрерывных распределений**

Выполнил:
Студент 1-го курса магистратуры
Малов И. М.

Группа: КММО-11-24

МОСКВА 2025

Задание 1. Моделирование показательного распределения.

Получить две выборки из $N = 200$ псевдослучайных чисел, распределенных по показательному закону с параметром λ:

1) используя метод обратной функции распределения и псевдослучайные числа, равномерно распределенные на интервале $(0,1)$;

2) используя одну из функций Python, например, numpy.random.exponential(1/**λ**, *N*).

Полученные выборки упорядочить по возрастанию, построить по ним группированные выборки.

Проверить при уровне значимости $\alpha = 0{,}05$ гипотезы о соответствии каждой выборки теоретическому распределению.

Задание 2. Моделирование гиперпоказательного распределения.

Получить выборку из $N = 200$ псевдослучайных чисел, распределенных по гиперпоказательному закону с параметрами $(\lambda_1, \lambda_2, \lambda_3, q_1, q_2, q_3)$, используя метод дискретной суперпозиции, псевдослучайные числа, равномерно распределенные на интервале $(0,1)$ и формулы из лекций.

Полученную выборку упорядочить по возрастанию, построить по ней группированную выборку в форме таблицы 1 из **Указания**.

Проверить при уровне значимости $\alpha = 0{,}05$ гипотезу о соответствии выборки теоретическому распределению.

Краткие теоретические сведения

В **Задании 1** рассматриваем показательное распределение:

функция распределения $F(x)=\begin{cases}0, x\le 0;\\ 1-e^{-\lambda x}, x>0;\end{cases}$

плотность распределения $f(x)=\begin{cases}0, x<0;\\ \lambda e^{-\lambda x}, x\ge 0;\end{cases}$

математическое ожидание $\frac{1}{\lambda}$;

дисперсия $\frac{1}{\lambda^2}$.

метод обратной функции:

$$G(y)=\frac{\ln(1-y)}{-\lambda}$$

Выборка создается следующим алгоритмом:

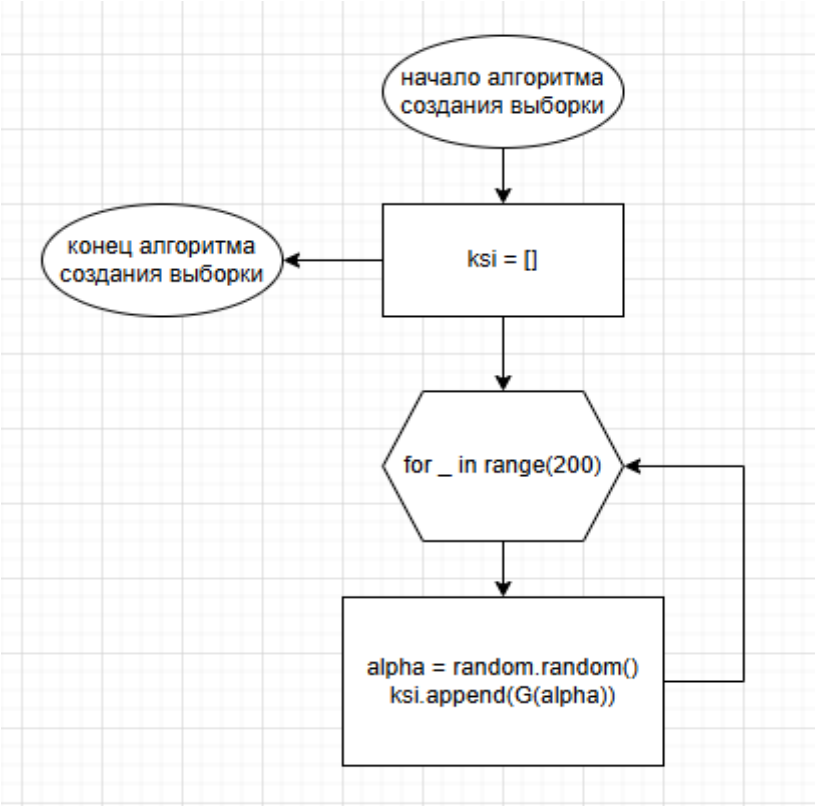

Рисунок 1. Алгоритм выборки от обратной функции

В **Задании 2** рассматриваем гиперпоказательное распределение:

функция распределения $F(x)=\begin{cases}0, x\le 0;\\ 1-q_1 e^{-\lambda_1 x}-q_2 e^{-\lambda_2 x}-q_3 e^{-\lambda_3 x}, x>0;\end{cases}$

$\lambda_i>0$, $q_i>0$, $q_1+q_2+q_3=1$;

плотность распределения $f(x)=\begin{cases}0, x<0;\\ q_1\lambda_1 e^{-\lambda_1 x}+q_2\lambda_2 e^{-\lambda_2 x}+q_3\lambda_3 e^{-\lambda_3 x}, x\geq 0;\end{cases}$

математическое ожидание $\frac{q_1}{\lambda_1}+\frac{q_2}{\lambda_2}+\frac{q_3}{\lambda_3}$

дисперсия $\frac{q_1}{\lambda_1^2}+\frac{q_1}{\lambda_1^2}+\frac{q_3}{\lambda_3}$

Обратные функции для метода дискретной суперпозиции:

$$G_1(y)=\frac{\ln(1-y)}{-\lambda_1}$$

$$G_2(y)=\frac{\ln(1-y)}{-\lambda_2}$$

$$G_3(y)=\frac{\ln(1-y)}{-\lambda_3}$$

Выборка создается следующим алгоритмом:

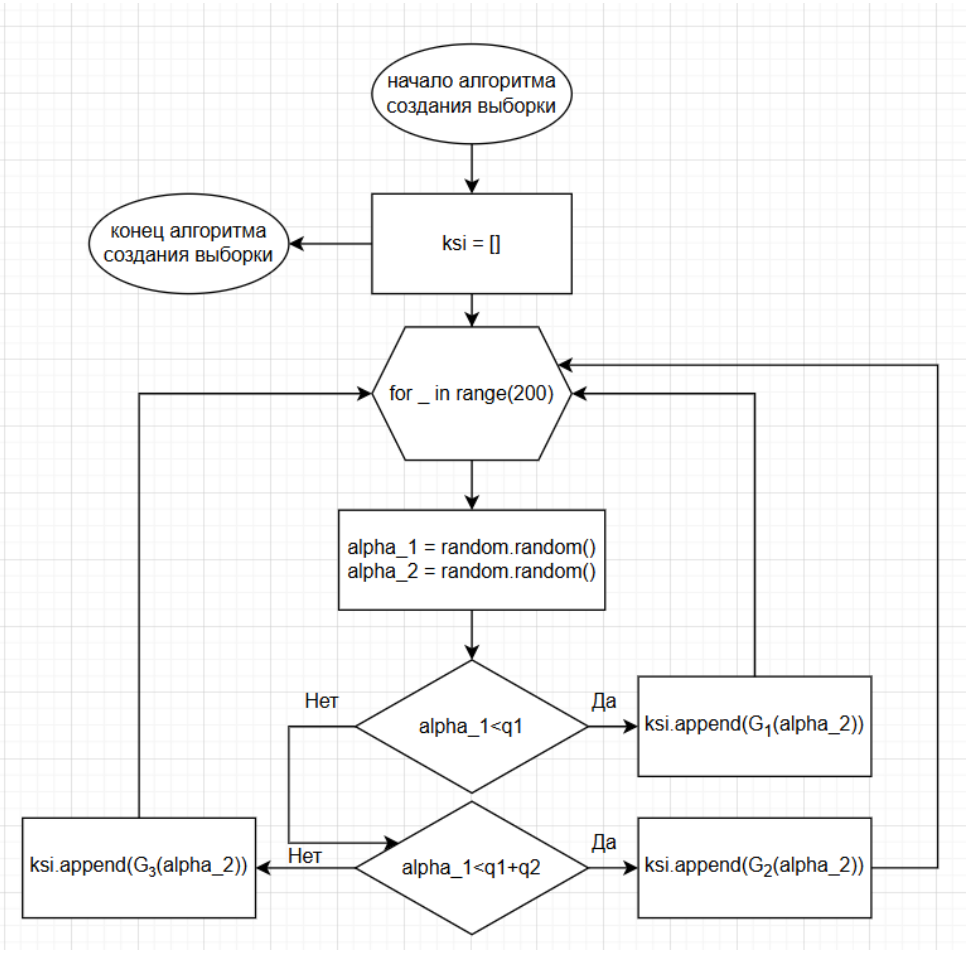

Рисунок 2. Алгоритм выборки от метода дискретной суперпозиции

Результаты расчетов

Вариант 8
Задание 1

lambda = 0.81

Данные, полученные с помощью обратной функции:

1.04597	1.42953	0.21000	0.65997	1.07939	0.66227	0.20592	1.36872	0.55934	0.37779
0.69157	1.33859	0.31357	0.34775	0.61143	0.04439	0.33450	0.18803	1.21839	0.71946
1.06539	0.17616	0.65742	0.39651	2.51448	0.00350	1.19167	1.11629	1.21793	0.51584
0.28493	1.80286	0.04645	1.18565	1.38133	0.94644	1.43409	3.11171	0.29038	1.60077
2.07206	4.98841	0.84738	0.24269	0.34110	0.06407	1.27025	0.26316	1.16176	0.81350
2.14190	4.30416	2.10328	0.15709	0.91998	0.12016	1.09632	0.53158	2.56651	0.73083
1.30839	1.17306	1.32423	0.75101	2.98012	0.14215	2.18342	0.10260	2.53156	0.80023
0.21559	0.05589	0.94131	0.22253	0.90642	0.74726	0.80481	0.26727	1.30409	0.80711
0.90786	0.00502	2.38505	1.29159	1.13774	6.07191	2.03472	0.28743	0.08982	1.44447
0.49032	0.17758	0.20019	2.12430	0.08004	0.81938	4.04886	0.12460	0.68001	1.57089
0.35516	3.49414	1.03403	1.85541	0.82980	0.75762	0.70563	10.00643	0.11511	1.99375
3.77002	0.44539	0.57862	0.80763	0.76402	0.70816	2.41716	2.69766	0.07017	0.64151
7.00174	0.56288	1.13526	0.52723	0.63298	0.85849	1.56384	0.76935	0.36292	1.94771
0.05627	2.82063	0.14792	0.70422	0.67343	0.73155	2.71032	1.68076	0.75360	4.12915
2.42872	0.46594	1.84437	0.00620	1.08558	1.59928	0.05844	1.46694	1.75879	6.01602
1.14111	0.98302	0.12543	1.53924	0.95511	2.81754	3.01603	0.91246	1.22361	1.57446
0.59337	0.70711	0.22464	0.49747	0.83104	1.47765	1.79756	0.67995	0.40983	0.08362
0.41178	0.08297	2.03382	0.47508	0.22380	0.67957	4.04962	1.34674	1.46626	2.31999
1.38743	1.08442	3.64277	0.10284	0.71880	1.29442	4.91391	1.00105	0.87191	2.23541
0.75350	2.29424	0.70099	0.73371	4.27585	2.96649	0.41500	0.38160	0.87447	0.02031

Отсортированные данные, полученные с помощью обратной функции:

0.00350	0.00502	0.00620	0.02031	0.04439	0.04645	0.05589	0.05627	0.05844	0.06407
0.07017	0.08004	0.08297	0.08362	0.08982	0.10260	0.10284	0.11511	0.12016	0.12460
0.12543	0.14215	0.14792	0.15709	0.17616	0.17758	0.18803	0.20019	0.20592	0.21000
0.21559	0.22253	0.22380	0.22464	0.24269	0.26316	0.26727	0.28493	0.28743	0.29038
0.31357	0.33450	0.34110	0.34775	0.35516	0.36292	0.37779	0.38160	0.39651	0.40983
0.41178	0.41500	0.44539	0.46594	0.47508	0.49032	0.49747	0.51584	0.52733	0.53158
0.55934	0.56288	0.57862	0.59337	0.61143	0.63298	0.64151	0.65742	0.65997	0.66227
0.67343	0.67957	0.67995	0.68001	0.69157	0.70099	0.70422	0.70563	0.70711	0.70816
0.71880	0.71946	0.73083	0.73155	0.73371	0.74726	0.75101	0.75350	0.75360	0.75762
0.76402	0.76935	0.80023	0.80481	0.80711	0.80763	0.81350	0.81938	0.82980	0.83104
0.84738	0.85849	0.87191	0.87447	0.90642	0.90786	0.91246	0.91998	0.94131	0.94644
0.95511	0.98302	1.00105	1.03403	1.04597	1.06539	1.07939	1.08442	1.08558	1.09632
1.11629	1.13526	1.13774	1.14111	1.16176	1.17306	1.18565	1.19167	1.21793	1.21839
1.22361	1.27025	1.29159	1.29442	1.30409	1.30839	1.32423	1.33859	1.34674	1.36872
1.38133	1.38743	1.42953	1.43409	1.44447	1.46626	1.46694	1.47765	1.53924	1.56384
1.57089	1.57446	1.59928	1.60077	1.68076	1.75879	1.79756	1.80286	1.84437	1.85541
1.94771	1.99375	2.03382	2.03472	2.07206	2.10328	2.12430	2.14190	2.18342	2.23541
2.29424	2.31999	2.38505	2.41716	2.42872	2.51448	2.53156	2.56651	2.69766	2.71032
2.81754	2.82063	2.96649	2.98012	3.01603	3.11171	3.49414	3.64277	3.77002	4.04886
4.04962	4.12915	4.27585	4.30416	4.91391	4.98841	6.01602	6.07191	7.00174	10.00643

Данные, полученные с помощью np.random.exponential:

0.20670	1.78556	0.63195	1.04886	1.25236	0.60060	0.26276	4.27350	1.17410	3.14669
1.66329	0.02596	6.57573	2.00886	2.44923	1.11815	0.635988	2.55329	1.18934	2.64434
0.37728	0.17934	0.24134	0.14888	3.50751	1.16504	1.908313	3.01157	0.01660	0.06468
0.94199	0.15260	4.05679	0.40063	0.72829	1.13005	0.361448	0.22989	1.05473	1.84122
0.01813	0.45792	0.67091	1.26707	1.36856	0.46220	0.143155	0.47035	1.85634	0.13819
3.09900	1.37417	0.68116	0.83793	3.18689	0.08943	3.295018	0.34171	3.69553	2.91118
2.85156	0.78667	0.76794	0.87203	1.19268	1.99150	0.166812	0.83360	1.49872	1.09439
0.04197	2.09103	0.57773	0.76402	0.48083	2.99260	1.146769	0.96931	0.15251	5.46958
3.88318	0.42211	0.77338	2.77945	0.47518	0.26730	0.423040	0.64192	0.34958	0.69108
0.18220	1.63084	0.04420	1.14919	0.42536	0.29197	1.215946	1.67455	0.67025	1.22697
0.41214	1.49723	0.10869	1.14162	3.89668	0.58508	1.329962	1.27417	1.23427	1.84860
1.15014	0.48995	1.62804	0.72439	0.64297	0.63652	0.086579	1.32948	0.12388	0.37143
3.56350	0.50301	1.24834	0.80743	3.61620	2.67206	0.542212	0.19951	0.61087	1.25758
2.36485	4.71526	0.03494	2.70773	2.40277	0.27882	0.503152	1.69472	0.47970	2.16458
0.00279	1.20953	0.44064	0.28867	1.44164	0.67041	0.241253	1.26627	4.38486	0.90950
0.90929	3.70620	0.30809	3.57141	0.00407	1.28890	0.009920	0.43722	4.29589	0.68209
0.99142	1.80094	0.06987	0.09421	2.84398	0.88483	0.012955	0.58189	1.67939	0.93476
0.82015	2.15188	0.91451	1.11635	3.11610	0.50150	1.123405	5.52665	0.90021	3.60292
1.80444	0.64439	0.66480	0.03907	0.00732	1.86076	0.109181	0.52161	0.22185	8.92283
0.21631	0.74101	0.06101	1.35110	1.27137	0.17967	0.468287	0.51944	3.68326	1.37218

Отсортированные данные, полученные с помощью np.random.exponential:

0.00279	0.00407	0.00738	0.00992	0.01296	0.01660	0.01813	0.02596	0.03494	0.03907
0.04197	0.04420	0.06101	0.06468	0.06987	0.08658	0.08943	0.09421	0.10869	0.10918
0.12388	0.13819	0.14316	0.14888	0.15251	0.15260	0.16681	0.17934	0.17967	0.18220
0.19951	0.20670	0.21631	0.22185	0.22989	0.24125	0.24134	0.26276	0.26730	0.27882
0.28867	0.29197	0.30809	0.34171	0.34958	0.36145	0.37143	0.37728	0.40063	0.41214
0.42211	0.42304	0.42536	0.43722	0.44064	0.45792	0.46220	0.46829	0.47035	0.47518
0.47970	0.48083	0.48995	0.50150	0.50301	0.50315	0.51944	0.52161	0.54221	0.57773
0.58189	0.58508	0.60060	0.61087	0.63195	0.63599	0.63652	0.64192	0.64297	0.64439
0.66480	0.67025	0.67041	0.67091	0.68116	0.68209	0.69108	0.72439	0.72829	0.74101
0.76402	0.76794	0.77338	0.78667	0.80743	0.82015	0.83360	0.83793	0.87203	0.88483
0.90021	0.90929	0.90950	0.91451	0.93476	0.94199	0.96931	0.99142	1.04886	1.05473
1.09439	1.11635	1.11815	1.12341	1.13005	1.14162	1.14677	1.14919	1.15014	1.16504
1.17410	1.18934	1.19268	1.20953	1.21595	1.22697	1.23427	1.24834	1.25236	1.25758
1.26627	1.26707	1.27137	1.27417	1.28890	1.32948	1.32996	1.35110	1.36856	1.37218
1.37417	1.44164	1.49723	1.49872	1.62804	1.63084	1.66329	1.67455	1.67455	1.69472
1.78556	1.80094	1.80444	1.84122	1.84860	1.85634	1.86076	1.90831	1.99150	2.00886
2.09103	2.15188	2.16458	2.36485	2.40277	2.44923	2.55329	2.64434	2.67206	2.70773
2.77945	2.84398	2.85156	2.91118	2.99260	3.01157	3.09900	3.11610	3.14669	3.18689
3.29502	3.50751	3.56350	3.57141	3.60292	3.61620	3.68326	3.69553	3.70620	3.88318
3.89668	4.05679	4.27350	4.29589	4.38486	4.71526	5.46958	5.52665	6.57573	8.92283

Сравнение относительных частот данных, полученных из обратной функции:

Интервал	n_i	w_i	p_i	$\lvert w_i - p_i \rvert$
[0, 1.25080]	131	0.655	0.63693	0.01807
(1.25080, 2.50161]	44	0.22	0.23125	0.01125
(2.50161, 3.75241]	13	0.065	0.08396	0.01896
(3.75241, 5.00322]	8	0.04	0.03048	0.00952
(5.00322, 6.25402]	2	0.01	0.01107	0.00107
(6.25402, 7.50482]	1	0.005	0.00402	0.00098
(7.50482, 8.75563]	0	0.0	0.00146	0.00146
(8.75563, 10.00643]	1	0.005	0.00053	0.00447
	200	1.0	0.9997	0.01896

Рисунок 1. Графики $w\_i\_ksi(w_i)$ и p_i

Сравнение относительных частот данных, полученных из numpy:

Интервал	n_i	w_i	p_i	$\lvert w_i - p_i \rvert$
[0, 1.11535]	111	0.555	0.59482	0.03982
(1.11535, 2.23071]	52	0.26	0.24101	0.01899
(2.23071, 3.34606]	18	0.09	0.09765	0.00765
(3.34606, 4.46142]	14	0.07	0.03957	0.03043
(4.46142, 5.57677]	3	0.015	0.01603	0.00103
(5.57677, 6.69213]	1	0.005	0.00650	0.0015
(6.69213, 7.80748]	0	0.0	0.00263	0.00263
(7.80748, 8.92283]	1	0.005	0.00107	0.00393
	200	1.0	0.99928	0.03982

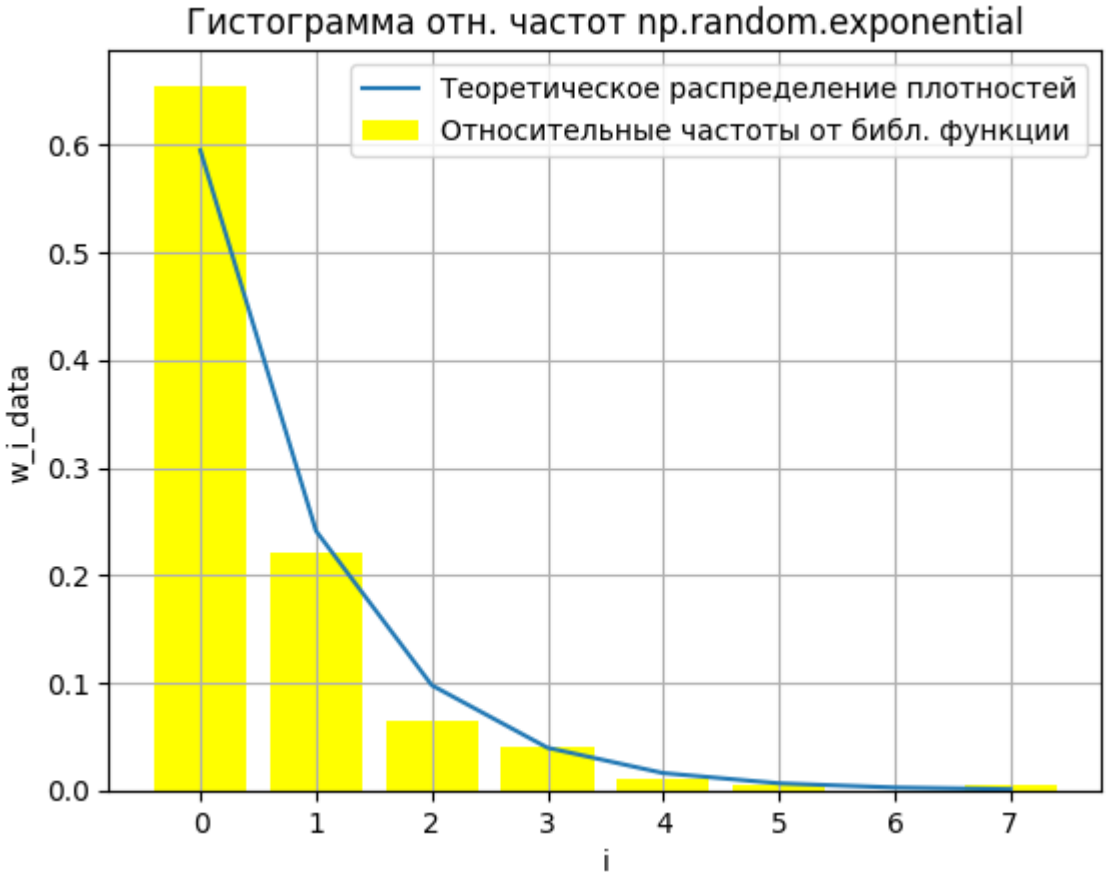

Рисунок 2. Графики $w\_i\_data(w_i)$ и p_i

Расчет хи-квадрат для смоделированной выборки:

i	a_i	F(a_i)	w_i	p_i	$\frac{N(w_i - p_i)^2}{p_i}$
0	0	0	–	–	–
1	1.25080	0.63693	0.655	0.63693	0.10253
2	2.50161	0.86818	0.22	0.23125	0.10946
3	3.75241	0.95214	0.065	0.08396	0.85632
4	5.00322	0.98262	0.04	0.03048	0.59469
5	6.25402	0.99369	0.01	0.01107	0.02068
6	7.50482	0.99771	0.005	0.00402	0.04778
7	8.75563	0.99917	0.0	0.00146	0.292
8	10.00643	0.9997	0.005	0.00053	7.53996
			1.0	0.9997	9.56342

Значение хи-квадрат смоделированной выборки: 9.56342

Критическое значение: 14.06714

Вывод: Не можем отвергнуть нулевую гипотезу. Данные согласуются с показательным распределением.

Расчет хи-квадрат для выборки, полученной с помощью numpy:

i	a_i	F(a_i)	w_i	p_i	$\frac{N(w_i - p_i)^2}{p_i}$
0	0	0	–	–	–
1	1.11535	0.59482	0.555	0.59482	0.53315
2	2.23071	0.83583	0.26	0.24101	0.29926
3	3.34606	0.93348	0.09	0.09765	0.11986
4	4.46142	0.97305	0.07	0.03957	4.68024
5	5.57677	0.98908	0.015	0.01603	0.01324
6	6.69213	0.99558	0.005	0.00650	0.06923
7	7.80748	0.99821	0.0	0.00263	0.526
8	8.92283	0.99928	0.005	0.00107	2.8869
			1.0	0.99928	9.12788

Значение хи-квадрат выборки, полученной с помощью numpy: 9.12788

Критическое значение: 14.06714

Вывод: Не можем отвергнуть нулевую гипотезу. Данные согласуются с показательным распределением

Задание 2

$$\lambda_1 = 0.55, \lambda_2 = 1.87, \lambda_3 = 0.32, \qquad q_1 = 0.34, q_2 = 0.43, q_3 = 0.23$$

Данные, полученные с помощью ДСП:

3.41847	0.15747	0.3192	0.83133	0.05229	1.77183	0.00113	0.68368	5.28247	0.09709
1.41746	1.49994	0.99706	0.4557	0.36621	0.47726	1.119	0.0395	2.54048	0.10999
6.14291	0.73547	0.66417	1.71365	3.46747	1.61039	5.85027	0.47549	0.50554	1.67179
1.44699	0.60019	0.30907	0.80592	0.26719	0.01605	2.20966	2.81061	0.09987	2.15062
2.42405	0.82561	1.62906	0.32865	1.44375	0.01789	5.36719	4.21566	2.28506	0.29694
3.70963	0.11922	0.35478	0.22715	0.52236	0.32768	0.47428	1.59669	5.0052	5.2975
3.3519	0.04647	0.02667	0.15353	1.78237	0.03853	1.36266	0.57947	0.32429	1.16731
0.88849	0.11139	4.18322	0.12544	0.90828	0.03277	2.30694	2.21717	0.42075	6.75081
2.27159	1.90103	0.45864	1.8124	0.64488	6.51164	3.55556	5.09386	0.28778	0.01548
0.12079	10.23586	0.10062	11.37828	0.38024	0.22785	2.03878	0.11541	1.99361	1.25954
3.73231	0.05706	0.72202	4.77024	0.43557	0.97236	1.31677	3.63095	2.26426	2.35032
0.33811	0.7484	0.49027	1.43449	0.18407	2.89468	6.82099	1.45657	0.07259	3.65668
5.51002	0.51069	1.484	0.13542	0.56406	2.59731	1.9127	2.81188	0.03893	0.05142
3.93336	0.4981	1.70774	0.33306	2.47746	4.39359	1.87989	1.08804	1.17853	1.91945
0.0192	1.09481	0.31955	0.11824	10.9386	0.36896	1.74878	2.44651	0.5798	1.41821
0.51431	0.59927	1.85634	0.55151	1.54713	0.72067	3.41491	1.12718	1.42865	0.48462
0.03743	0.1365	0.03934	1.83437	2.85656	1.06161	1.26547	0.25076	0.59451	13.10246
1.2042	1.31324	1.08568	0.47743	0.10086	4.24355	7.42753	4.28484	0.10492	0.0919
0.58928	0.12106	5.16118	0.41686	1.8871	0.22112	0.04493	0.28825	0.03441	0.07959
0.54529	1.58488	0.5069	0.01365	0.37163	0.30846	0.703	1.24244	0.13635	0.43952

Отсортированные данные ДСП:

0.00113	0.01365	0.01548	0.01605	0.01789	0.0192	0.02667	0.03277	0.03441	0.03743
0.03853	0.03893	0.03934	0.0395	0.04493	0.04647	0.05142	0.05229	0.05706	0.07259
0.07959	0.0919	0.09709	0.09987	0.10062	0.10086	0.10492	0.10999	0.11139	0.11541
0.11824	0.11922	0.12079	0.12106	0.12544	0.13542	0.13635	0.1365	0.15353	0.15747
0.18407	0.22112	0.22715	0.22785	0.25076	0.26719	0.28778	0.28825	0.29694	0.30846
0.30907	0.3192	0.31955	0.32429	0.32768	0.32865	0.33306	0.33811	0.35478	0.36621
0.36896	0.37163	0.38024	0.41686	0.42075	0.43557	0.43952	0.4557	0.45864	0.47428
0.47549	0.47726	0.47743	0.48462	0.49027	0.4981	0.50554	0.5069	0.51069	0.51431
0.52236	0.54529	0.55151	0.56406	0.57947	0.5798	0.58928	0.59451	0.59927	0.60019
0.64488	0.66417	0.68368	0.703	0.72067	0.72202	0.73547	0.7484	0.80592	0.82561
0.83133	0.88849	0.90828	0.97236	0.99706	1.06161	1.08568	1.08804	1.09481	1.119
1.12718	1.16731	1.17853	1.2042	1.24244	1.25954	1.26547	1.31324	1.31677	1.36266
1.41746	1.41821	1.42865	1.43449	1.44375	1.44699	1.45657	1.484	1.49994	1.54713
1.58488	1.59669	1.61039	1.62906	1.67179	1.70774	1.71365	1.74878	1.77183	1.78237
1.8124	1.83437	1.85634	1.87989	1.8871	1.90103	1.9127	1.91945	1.99361	2.03878
2.15062	2.20966	2.21717	2.26426	2.27159	2.28506	2.30694	2.35032	2.42405	2.44651
2.47746	2.54048	2.59731	2.81061	2.81188	2.85656	2.89468	3.3519	3.41491	3.41847
3.46747	3.55556	3.63095	3.65668	3.70963	3.73231	3.93336	4.18322	4.21566	4.24355
4.28484	4.39359	4.77024	5.0052	5.09386	5.16118	5.28247	5.2975	5.36719	5.51002
5.85027	6.14291	6.51164	6.75081	6.82099	7.42753	10.23586	10.9386	11.37828	13.10246

Сравнение относительных частот данных, полученных с помощью ДСП:

Интервал	n_i	w_i	p_i	$\lvert w_i - p_i \rvert$
[0, 1.63781]	134	0.67	0.70559	0.03559
(1.63781, 3.27562]	33	0.165	0.15673	0.00827
(2.91166, 4.91342]	16	0.08	0.0671	0.0129
(4.91342, 6.55123]	10	0.05	0.03305	0.01695
(6.55123, 8.18904]	3	0.015	0.01703	0.00203
(8.18904, 9.82685]	0	0.0	0.00906	0.00906
(9.82685, 11.46465]	3	0.015	0.00495	0.01005
(11.46465, 13.10246]	1	0.005	0.00276	0.00224
	200	1.0	0.99627	0.03559

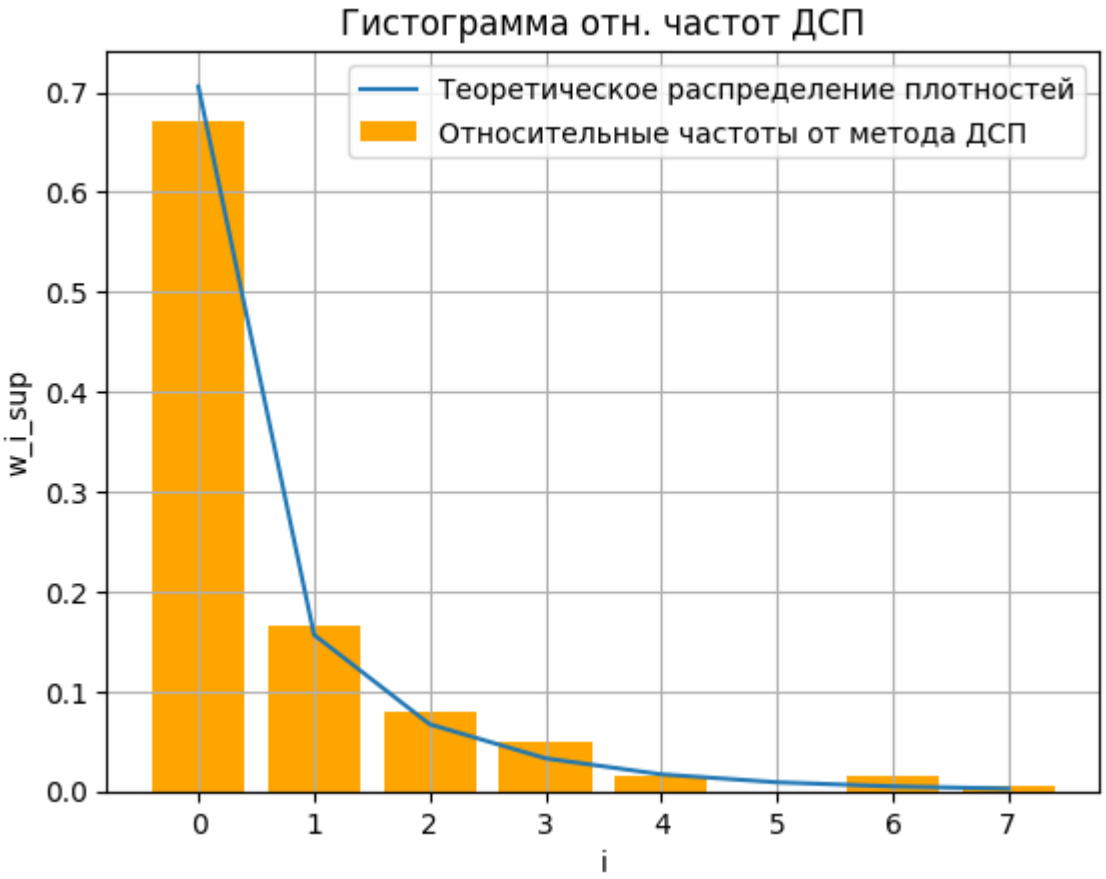

Рисунок 4. Графики $w\_i\_sup(w_i)$ и p_i

Расчет хи-квадрат для выборки ДСП:

i	a_i	F(a_i)	w_i	p_i	$\frac{N(w_i - p_i)^2}{p_i}$
0	0	0	–	–	–
1	1.63781	0.70559	0.67	0.70559	0.35903
2	3.27562	0.86232	0.165	0.15673	0.08727
3	4.91342	0.92942	0.08	0.0671	0.49601
4	6.55123	0.96247	0.05	0.03305	1.73859
5	8.18904	0.9795	0.015	0.01703	0.0484
6	9.82685	0.98856	0.0	0.00906	1.812
7	11.46465	0.99351	0.015	0.00495	4.08091
8	13.10246	0.99627	0.005	0.00276	0.36359
			1.0	0.99627	8.98580

Значение хи-квадрат смоделированной выборки: 8.98580

Критическое значение: 14.06714

Вывод: Не можем отвергнуть нулевую гипотезу. Данные согласуются с гиперпоказательным распределением.

Список литературы

1. Лобузов А.А. Статистическое моделирование [Электронный ресурс]: методические указания. – М.: МИРЭА – Российский технологический университет, 2023.
2. Ермаков С.М., Михайлов Г.А. Статистическое моделирование. – М.: Наука, 1982 г. – 296 с.
3. Соболь И.М. Численные методы Монте-Карло. – М.: Наука, 1973 г. – 312 с.
4. Бусленко Н.П., Голенко Д. И., Соболь И. М., Срагович В. Г., Шрейдер Ю.А. Метод статистических испытаний (метод Монте-Карло). – М.: Гос. изд-во физико-математической литературы, 1962 г. – 332 с.

Приложение

```
#Задание1

import numpy as np
import matplotlib.pyplot as plt
import math
import random
import scipy.stats
#8 вариант

lamb = 0.81

def f_exp(x):
    return lamb*math.exp(-(lamb*x))
def my_exp(x):
    return 1-math.exp(-lamb*x)
def reverse_exp(y):
    return round(-math.log(1-y)/(lamb),8)

def generate_ksi_exp(size = 200):
    ksi = []
    for _ in range(size):
        alpha = random.random()
        ksi.append(round(reverse_exp(alpha),5))
    return ksi

random.seed(10)

ksi_exp = generate_ksi_exp()
ksi_exp = np.array(ksi_exp)
output_ksi = "\n".join(map('{:.5f}'.format, ksi_exp))
print(f"Сгенерированная выборка САМНР: \n {output_ksi}")
ksi_exp.sort()
print(ksi_exp)

output_ksi = "\n".join(map('{:.5f}'.format, ksi_exp))
print(f"\n Отсортированная выборка САМНР: \n {output_ksi}\n")

ksi_exp_E_x = sum(ksi_exp)/200
print(f"Матожидание выборки САМНР:  {ksi_exp_E_x}")
squares = [(x-ksi_exp_E_x)**2 for x in ksi_exp]
ksi_exp_D_x = sum(squares)/199
print(f"Дисперсия выборки САМНР: {ksi_exp_D_x }")

np.random.seed(12)

data_exp = np.random.exponential(1/lamb,200)
output_data = "\n".join(map('{:.6f}'.format, data_exp))
print(f"Сгенерированная выборка numpy: \n {output_data}")
data_exp.sort()
output_data = "\n".join(map('{:.6f}'.format, data_exp))
print(f"\n Отсортированная выборка numpy: \n {output_data}\n")

data_exp_E_x = sum(data_exp)/200
print(f"Матожидание выборки САМНР: {data_exp_E_x}")
squares = [(x-data_exp_E_x)**2 for x in data_exp]
data_exp_D_x = sum(squares)/199
print(f"Дисперсия выборки САМНР: {data_exp_D_x}")

print("KSI_EXP")
```

```
m = 8
a_0 = 0
a_m = max(ksi_exp)
interval = (a_m - a_0)/m
a_i_ksi = [round(a_0 + i*interval,7) for i in range(1,m+1)]
print(a_i_ksi)
n_i = np.zeros(len(a_i_ksi))
count = 0
for ksi in ksi_exp:
    i = 0
    while ksi > a_i_ksi[i]:
        i+=1
    n_i[i]+=1
w_i_ksi = [round(n_i[i]/200,7) for i in range(len(n_i))]
print(n_i)
print(w_i_ksi,"\n")


print("DATA_EXP")
a_0 = 0
a_m = max(data_exp)
interval = (a_m - a_0)/m
a_i_data = [round(a_0 + i*interval,7) for i in range(1,m+1)]
print(a_i_data)
n_i = np.zeros(len(a_i_data))
count = 0
for ksi in data_exp:
    i = 0
    while ksi > a_i_data[i]:
        i+=1
    n_i[i]+=1
w_i_data = [n_i[i]/200 for i in range(len(n_i))]
print(n_i)
print(w_i_data)

prob = [round(my_exp(i+1)-my_exp(i),7) for i in range(len(w_i_ksi))]
print(prob)
print(sum(prob))

x1 = [k for k in range(len(w_i_ksi))]
y1 = [s for s in w_i_ksi]
x2 = [k for k in range(len(w_i_data))]
y2 = [s for s in w_i_data]
x3 = [k for k in range(len(prob))]
y3 = [s for s in prob]

plt.plot(x1,y1,marker ='o',label='САМНР', color = 'red')
plt.plot(x2,y2,marker ='o',label='np.exp', color = 'green')
plt.plot(x3,y3,marker ='o',label='Теор.зн', color = 'blue')
plt.title("Показательное распределение")
plt.xlabel("Промежутки")
plt.ylabel("Относительная частота")
plt.legend()
plt.grid()
plt.show()

ksi_exp_dif = [round(w_i_ksi[i]-prob[i],7) for i in range(len(w_i_ksi))]
print(ksi_exp_dif)
data_exp_dif = [round(w_i_data[i]-prob[i],7) for i in range(len(w_i_data))]
print(data_exp_dif)


hi_ksi = [200*ksi_exp_dif[i]**2/prob[i] for i in range(len(ksi_exp_dif))]
print(hi_ksi)
```

```
hi_data = [200*data_exp_dif[i]**2/prob[i] for i in range(len(data_exp_dif))]
print(hi_data)

print(f"Хи-квадрат САМНР {sum(hi_ksi)}")
print(f"Хи-квадрат numpy {sum(hi_data)}")
scipy.stats.chi2.ppf(0.95,m-1)

plt.bar([x for x in range(len(w_i_ksi))],w_i_ksi,color = 'red',label =
'Относительные частоты от обратной функции')
plt.plot([x for x in range(len(prob))],prob,label = 'Теоретическое
распределение плотностей')

plt.title("Гистограмма отн. частот обратной функции")
plt.xlabel("i")
plt.ylabel("w_i_ksi")
plt.grid()
plt.legend()
plt.show()

plt.bar([x for x in range(len(w_i_ksi))],w_i_ksi,color = 'Yellow',label =
'Относительные частоты от библ. функции')
plt.plot([x for x in range(len(prob))],prob,label = 'Теоретическое
распределение плотностей')

plt.title("Гистограмма отн. частот np.random.exponential")
plt.xlabel("i")
plt.ylabel("w_i_data")
plt.grid()
plt.legend()
plt.show()

for a in a_i_ksi:
    print(round(my_exp(a),5))

for a in a_i_data:
    print(round(my_exp(a),5))

#Задание2

import numpy as np
import matplotlib.pyplot as plt
import math
import random
import scipy.stats
from docx import Document

lambda1 = 0.55
lambda2 = 1.87
lambda3 = 0.32
q1 = 0.34
q2 = 0.43
q3 = 0.23

def f(x):
    return q1*lambda1*math.exp(-lambda1*x)+q2*lambda2*math.exp(-
lambda2*x)+q3*lambda3*math.exp(-lambda3*x)

def F(x):
    return 1 - q1*math.exp(-lambda1*x) - q2*math.exp(-lambda2*x) -
q3*math.exp(-lambda3*x)

def G_1(y):
    return math.log(1-y)/(-lambda1)
```

```
def G_2(y):
    return math.log(1 - y) / (-lambda2)
def G_3(y):
    return math.log(1-y)/(-lambda3)


def superpos():
    ksi= []
    for _ in range(200):
        alpha_1 = random.random()
        alpha_2 = random.random()
        if alpha_1<q1:
            ksi.append(round(G_1(alpha_2),5))
        elif alpha_1<q1+q2:
            ksi.append(round(G_2(alpha_2),5))
        else:
            ksi.append(round(G_3(alpha_2),5))
    return ksi

random.seed(1)
ksi_sup = superpos()
print(ksi_sup)

doc = Document()

doc.add_heading('Таблица случайных чисел (20x10)', level=1)

table = doc.add_table(rows=20, cols=10)

for i in range(20):
    for j in range(10):
        index = i * 10 + j
        table.cell(i, j).text = str(ksi_sup[index])

doc.save('случайные_числа.docx')

print("Документ создан и сохранен как 'случайные_числа.docx'")

ksi_sup.sort()
print(ksi_sup)

doc = Document()

doc.add_heading('Таблица случайных чисел (20x10)', level=1)

table = doc.add_table(rows=20, cols=10)

for i in range(20):
    for j in range(10):
        index = i * 10 + j
        table.cell(i, j).text = str(ksi_sup[index])

doc.save('случайные_числа.docx')

print("Документ создан и сохранен как 'случайные_числа.docx'")

print("ksi_sup")
m = 8
a_0 = 0
a_m = max(ksi_sup)
interval = (a_m - a_0)/m
a_i_sup = [round(a_0 + i*interval,7) for i in range(1,m+1)]
print(a_i_sup)
n_i = np.zeros(len(a_i_sup))
```

```
count = 0
for ksi in ksi_sup:
    i = 0
    while ksi > a_i_sup[i]:
        i+=1
    n_i[i]+=1
w_i_sup = [n_i[i]/200 for i in range(len(n_i))]
print(n_i)
print(w_i_sup)

prob = [round(F(i+1)-F(i),7) for i in range(len(w_i_sup))]
print(prob)
print(sum(prob))


x1 = [k for k in range(len(w_i_sup))]
y1 = [s for s in w_i_sup]
x2 = [k for k in range(len(prob))]
y2 = [s for s in prob]

plt.plot(x1,y1,marker ='o',label='Дискретная суперпозиция', color = 'red')
plt.plot(x2,y2,marker ='o',label='Теор. значения', color = 'green')
plt.title("Гиперпоказательное распределение")
plt.xlabel("Промежутки")
plt.ylabel("Относительная частота")
plt.legend()
plt.grid()
plt.show()

ksi_hyperexp_dif = [round(w_i_sup[i]-prob[i],7) for i in range(len(w_i_sup))]
print(ksi_hyperexp_dif)


hi_ksi = [200*ksi_hyperexp_dif[i]**2/prob[i] for i in
range(len(ksi_hyperexp_dif))]
print(hi_ksi)


print(f"Хи-квадрат ДСП {sum(hi_ksi)}")
scipy.stats.chi2.ppf(0.95,m-1)


plt.bar([x for x in range(len(w_i_sup))],w_i_sup,color = 'Orange',label =
'Относительные частоты от метода ДСП')
plt.plot([x for x in range(len(prob))],prob,label = 'Теоретическое
распределение плотностей')

plt.title("Гистограмма отн. частот ДСП")
plt.xlabel("i")
plt.ylabel("w_i_sup")
plt.grid()
plt.legend()
plt.show()

for a in a_i_sup:
    print(round(F(a),5))
```