



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

ИНСТИТУТ КИБЕРНЕТИКИ

КАФЕДРА ВЫСШЕЙ МАТЕМАТИКИ

## **Лабораторная работа 4**

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

Тема: **Проверка статистических гипотез с помощью критерия Колмогорова**

Выполнил:  
Студент 3-го курса  
Жолковскиц Д.А.

Группа: КМБО-01-16

МОСКВА 2019

**Лабораторная работа по Математической статистике № 4**  
**«Проверка статистических гипотез с помощью критерия**  
**Колмогорова»**

**Задание I. Проверка гипотезы о равномерном распределении.**

Из файла UD-2 в соответствии с номером варианта взять выборку  $\{x_1, \dots, x_N\}$  и значения  $a$  и  $b$ .

**Построить:**

на одном рисунке график эмпирической функции распределения этой выборки и график функции распределения равномерного закона на отрезке  $[a, b]$ .

**Проверить** в соответствии с Указаниями гипотезу о соответствии выборки равномерному распределению на отрезке  $[a, b]$  при уровне значимости 0,05 двумя способами – непосредственным расчетом и с помощью функции Python (`scipy.stats.kstest(sorted(array), 'uniform', (a, b))`).

**II. Проверка гипотезы об одинаковом распределении случайных величин.**

В соответствии с номером варианта взять выборку  $\{x_1, \dots, x_N\}$  из файла UD-2 и выборку  $\{y_1, \dots, y_M\}$  из файла UD-3.

**Построить:**

на одном рисунке графики эмпирических функций распределения этих выборок.

**Проверить** в соответствии с Указаниями гипотезу об одинаковом распределении двух случайных выборок при уровне значимости 0,05 двумя способами – непосредственным расчетом и с помощью функции Python (`scipy.stats.ks_2samp(arr_1, arr_2)`).

Графики должны быть отчётливыми: размер по горизонтали – по ширине страницы, по вертикали – не менее половины страницы; деления по горизонтали – через 1, по вертикали – через 0,1.

Вычисления проводить с точностью до 0,00001.

## Краткие теоретические сведения

### Равномерное распределение на отрезке $[a, b]$

Характеристика	Значение
Плотность	$f(x) = \begin{cases} 0, x \notin (a, b) \\ \frac{1}{b-a}, x \in (a, b) \end{cases}$
Функция распределения	$F(x) = \begin{cases} 0, x \leq a \\ \frac{x-a}{b-a}, x \in (a, b) \\ 1, x \geq b \end{cases}$
Математическое ожидание	$\frac{a+b}{2}$
Дисперсия	$\frac{(b-a)^2}{12}$
Среднее квадратичное отклонение	$\frac{b-a}{2\sqrt{3}}$

Пусть имеется случайная выборка  $\{X_1, \dots, X_N\}$  из непрерывного распределения с функцией распределения  $F(x)$ .

Эмпирическая функция распределения выборки

$$F_N^{\exists}(x; x_1, x_2, \dots, x_N) = \sum_{x_i \leq x} I_{(-\infty; x]}(x_j)$$

Статистика Колмогорова

$$D_N = \sup_{-\infty < x < +\infty} |F_N^{\exists}(x) - F(x)|$$

Теорема Колмогорова

Пусть  $F(x)$  – непрерывна. Тогда

$$P(D_N \sqrt{N} \leq z) \rightarrow_{N \rightarrow \infty} K(z) = \begin{cases} 0, z \leq 0 \\ \sum_{j=-\infty}^{+\infty} (-1)^j I^{-2j^2 z^2}, z > 0 \end{cases}$$

Общая схема проверки с помощью критерия Колмогорова статистических гипотез о соответствии выборки равномерному распределению

Функция распределения равномерного закона на отрезке  $[a, b]$

$$F(x) = \begin{cases} 0, x \leq a \\ \frac{x-a}{b-a}, x \in (a, b) \\ 1, x \geq b \end{cases}$$

Эмпирическая функция распределения выборки:

$$F_N^{\exists}(x; x_1, x_2, \dots, x_N) = \sum_{x_i \leq x} I_{(-\infty; x]}(x_j)$$

$$D_N = \max_{1 \leq j \leq N} (\max(|F_N(x_{(j)}) - F(x_{(j)})|, |F_N(x_{(j)} - 0) - F(x_{(j)})|))$$

Если  $D_N \sqrt{N} < k_{\alpha}$ , то гипотеза принимается.

Если  $D_N \sqrt{N} \geq k_{\alpha}$ , то гипотеза отвергается.

Критерий Колмогорова проверки гипотезы об одинаковом распределении наблюдаемых величин

$$D_{N,M} = \max_{j,k} \{ |F_N(x_{(j)}) - F_M(x_{(j)})|, |F_N(x_{(j)} - 0) - F_M(x_{(j)})|, \\ |F_N(y_{(k)}) - F_M(y_{(k)})|, |F_N(y_{(k)} - 0) - F_M(y_{(k)} - 0)| \};$$

$$K_{N,M} = D_{N,M} \sqrt{\frac{NM}{N+M}};$$

## **Средства языка программирования Python которые использованы в программе расчета**

В программе расчёта используются следующие средства языка:

Функции:

- `numpy.sort()` – сортирует вектор в порядке возрастания.
- `sorted(arr)` – возвращает отсортированную в порядке возрастания копию массива `arr`.
- `max(arr)` – возвращает максимальный элемент из массива `arr`.
- `min(arr)` – возвращает минимальный элемент из массива `arr`.
- `scipy.stats.kstest(array, 'uniform', (a, b)).pvalue` – проверка гипотезы о принадлежности выборки `data` распределению при помощи критерия Колмогорова-Смирнова.
- `scipy.stats.ks_2samp(arr_1, arr_2).pvalue` – проверка являются ли выборки из одного распределения.

Построение графиков и гистограмм было осуществлено с помощью библиотеки `matplotlib`. А построение таблиц было реализовано средствами библиотеки `docx`.

## Результаты расчетов

Задание 1)

$a = 2.0$

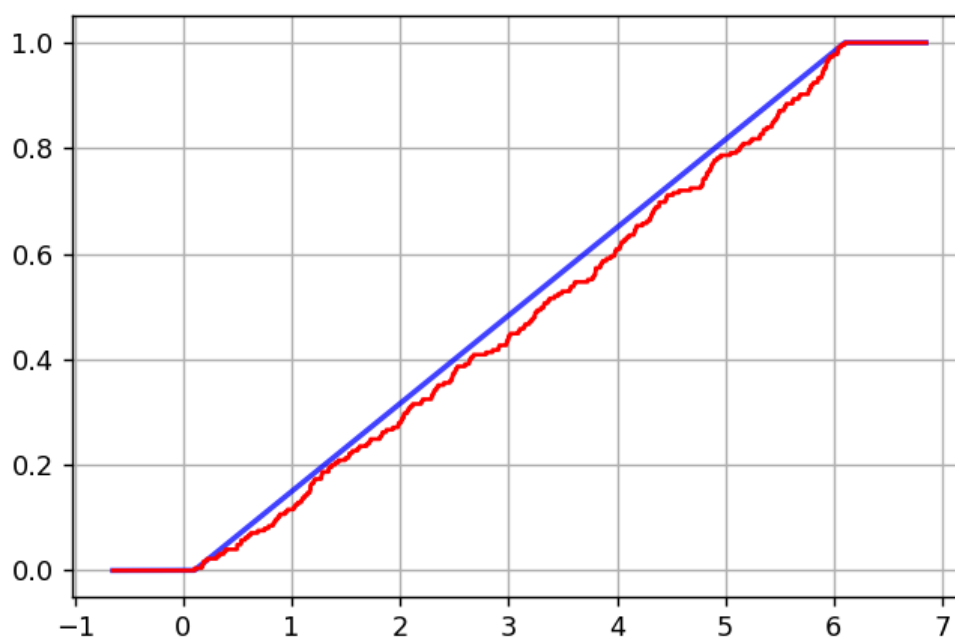
$b = 7.5$

Полученная выборка:

5.48848	3.01468	1.52398	4.17952	3.77314	4.15594	5.93128	0.78484	4.35748	5.76826
2.50318	0.49498	1.08742	5.12908	4.03246	2.48062	5.49532	4.83544	0.38986	5.56102
5.76352	4.31086	4.30144	5.81002	3.32386	2.99866	1.11922	1.98514	3.90286	3.85360
4.31812	3.99022	1.36768	5.90710	2.97988	0.49612	0.87382	3.18976	5.61922	4.17046
4.38838	5.46724	5.04448	3.97648	5.62390	1.01638	3.32800	4.87192	4.27534	5.93236
4.08688	2.29270	4.79428	4.57180	1.14088	5.92648	3.72838	5.78992	2.91190	5.95954
2.31490	5.22262	5.90020	4.33204	0.59668	5.44888	4.11088	3.56350	2.99590	4.16296
2.63944	5.99356	3.98080	2.48494	2.63464	2.33446	3.20950	4.81978	1.05436	5.42752
0.52954	0.82522	2.52532	2.90710	5.89408	2.09272	4.86634	2.06416	2.51224	0.93814
2.07454	1.33936	5.78236	2.03500	1.39936	0.10372	0.95908	3.61018	1.17760	5.48110
1.21030	3.78706	2.30752	4.46176	0.18784	2.67502	0.18010	0.30586	0.85420	2.79004
5.43328	4.33996	6.03304	1.42936	2.03332	4.88668	2.40160	4.79224	1.27930	5.68678
3.24352	3.22804	5.16154	5.94868	2.20612	2.28754	1.27138	1.93174	5.88634	1.68946
0.74812	1.72516	2.11474	3.94108	5.67454	4.82668	4.03888	5.54794	5.85418	4.06408
1.87168	0.62050	1.71376	3.24424	3.26290	4.23484	0.53296	4.45528	4.51744	4.44412
3.80044	0.16912	6.09958	4.38868	3.80344	1.19848	1.17004	1.17172	0.37240	1.81198
2.00848	3.13882	4.95424	4.85422	5.14078	2.59702	0.88810	1.83604	0.21754	2.34778
3.15160	0.68506	2.98336	1.62460	0.83896	2.45788	1.34152	1.55722	4.67602	2.47720
6.06682	2.01586	1.03162	3.49174	3.29734	5.84986	5.38360	5.34892	4.02508	3.57148
3.10180	4.89724	1.98040	2.19358	4.78906	6.03904	1.09852	1.49374	5.31676	3.60034
4.92538	3.80140	2.85154	3.36118	3.42166	3.86650	3.37708	6.03988	3.09022	5.24374
3.84316	0.57010	1.16368	4.77148	1.60462	1.53160	5.10202	1.27390	3.45700	5.35168
1.82686	0.32992	5.32156	2.65498	5.53840					

Упорядоченная выборка:

0.10372	0.16912	0.18010	0.18784	0.21754	0.30586	0.32992	0.37240	0.38986	0.49498
0.49612	0.52954	0.53296	0.57010	0.59668	0.62050	0.68506	0.74812	0.78484	0.82522
0.83896	0.85420	0.87382	0.88810	0.93814	0.95908	1.01638	1.03162	1.05436	1.08742
1.09852	1.11922	1.14088	1.16368	1.17004	1.17172	1.17760	1.19848	1.21030	1.27138
1.27390	1.27930	1.33936	1.34152	1.36768	1.39936	1.42936	1.49374	1.52398	1.53160
1.55722	1.60462	1.62460	1.68946	1.71376	1.72516	1.81198	1.82686	1.83604	1.87168
1.93174	1.98040	1.98514	2.00848	2.01586	2.03332	2.03500	2.06416	2.07454	2.09272
2.11474	2.19358	2.20612	2.28754	2.29270	2.30752	2.31490	2.33446	2.34778	2.40160
2.45788	2.47720	2.48062	2.48494	2.50318	2.51224	2.52532	2.59702	2.63464	2.63944
2.65498	2.67502	2.79004	2.85154	2.90710	2.91190	2.97988	2.98336	2.99590	2.99866
3.01468	3.09022	3.10180	3.13882	3.15160	3.18976	3.20950	3.22804	3.24352	3.24424
3.26290	3.29734	3.32386	3.32800	3.36118	3.37708	3.42166	3.45700	3.49174	3.56350
3.57148	3.60034	3.61018	3.72838	3.77314	3.78706	3.80044	3.80140	3.80344	3.84316
3.85360	3.86650	3.90286	3.94108	3.97648	3.98080	3.99022	4.02508	4.03246	4.03888
4.06408	4.08688	4.11088	4.15594	4.16296	4.17046	4.17952	4.23484	4.27534	4.30144
4.31086	4.31812	4.33204	4.33996	4.35748	4.38838	4.38868	4.44412	4.45528	4.46176
4.51744	4.57180	4.67602	4.77148	4.78906	4.79224	4.79428	4.81978	4.82668	4.83544
4.85422	4.86634	4.87192	4.88668	4.89724	4.92538	4.95424	5.04448	5.10202	5.12908
5.14078	5.16154	5.22262	5.24374	5.31676	5.32156	5.34892	5.35168	5.38360	5.42752
5.43328	5.44888	5.46724	5.48110	5.48848	5.49532	5.53840	5.54794	5.56102	5.61922
5.62390	5.67454	5.68678	5.76352	5.76826	5.78236	5.78992	5.81002	5.84986	5.85418
5.88634	5.89408	5.90020	5.90710	5.92648	5.93128	5.93236	5.94868	5.95954	5.99356
6.03304	6.03904	6.03988	6.06682	6.09958					



$a$	$b$	$N$	$D_N$	$D_N\sqrt{N}$	$x^*$	$F(x^*)$	$F_N(x^*)$	$F_N(x^* - 0)$
0.10000	6.10000	225	0.06108	0.91618	3.77314	0.61219	0.55556	0.55111

При втором способе проверки гипотезы о соответствии выборки равномерному распределению с помощью критерия Колмогорова значение функции:  $pval = 0.35840$

Задание 2)

Полученная выборка:

X

5.48848	3.01468	1.52398	4.17952	3.77314	4.15594	5.93128	0.78484	4.35748	5.76826
2.50318	0.49498	1.08742	5.12908	4.03246	2.48062	5.49532	4.83544	0.38986	5.56102
5.76352	4.31086	4.30144	5.81002	3.32386	2.99866	1.11922	1.98514	3.90286	3.85360
4.31812	3.99022	1.36768	5.90710	2.97988	0.49612	0.87382	3.18976	5.61922	4.17046
4.38838	5.46724	5.04448	3.97648	5.62390	1.01638	3.32800	4.87192	4.27534	5.93236
4.08688	2.29270	4.79428	4.57180	1.14088	5.92648	3.72838	5.78992	2.91190	5.95954
2.31490	5.22262	5.90020	4.33204	0.59668	5.44888	4.11088	3.56350	2.99590	4.16296
2.63944	5.99356	3.98080	2.48494	2.63464	2.33446	3.20950	4.81978	1.05436	5.42752
0.52954	0.82522	2.52532	2.90710	5.89408	2.09272	4.86634	2.06416	2.51224	0.93814
2.07454	1.33936	5.78236	2.03500	1.39936	0.10372	0.95908	3.61018	1.17760	5.48110
1.21030	3.78706	2.30752	4.46176	0.18784	2.67502	0.18010	0.30586	0.85420	2.79004
5.43328	4.33996	6.03304	1.42936	2.03332	4.88668	2.40160	4.79224	1.27930	5.68678
3.24352	3.22804	5.16154	5.94868	2.20612	2.28754	1.27138	1.93174	5.88634	1.68946
0.74812	1.72516	2.11474	3.94108	5.67454	4.82668	4.03888	5.54794	5.85418	4.06408
1.87168	0.62050	1.71376	3.24424	3.26290	4.23484	0.53296	4.45528	4.51744	4.44412
3.80044	0.16912	6.09958	4.38868	3.80344	1.19848	1.17004	1.17172	0.37240	1.81198
2.00848	3.13882	4.95424	4.85422	5.14078	2.59702	0.88810	1.83604	0.21754	2.34778
3.15160	0.68506	2.98336	1.62460	0.83896	2.45788	1.34152	1.55722	4.67602	2.47720
6.06682	2.01586	1.03162	3.49174	3.29734	5.84986	5.38360	5.34892	4.02508	3.57148
3.10180	4.89724	1.98040	2.19358	4.78906	6.03904	1.09852	1.49374	5.31676	3.60034

4.92538	3.80140	2.85154	3.36118	3.42166	3.86650	3.37708	6.03988	3.09022	5.24374
3.84316	0.57010	1.16368	4.77148	1.60462	1.53160	5.10202	1.27390	3.45700	5.35168
1.82686	0.32992	5.32156	2.65498	5.53840					

Y

1.45028	3.67544	0.99338	0.68942	3.77204	0.95696	5.69132	2.06816	4.84250	3.48920
1.37864	4.82846	0.30458	1.81586	0.23426	0.22784	1.00280	2.72180	0.31778	3.74090
5.51726	3.97742	4.89818	3.08882	4.89188	1.30232	1.86506	2.21042	5.53082	5.29076
4.22588	1.07132	0.44462	2.20448	1.00220	1.13900	3.04820	0.26750	4.98080	2.13080
0.27560	3.76202	1.00880	5.41478	4.74554	5.75366	6.06098	4.30694	3.34646	1.28078
5.75636	0.49358	5.30564	3.80342	1.94120	5.47946	4.25822	5.25854	2.12486	3.29726
4.58864	3.85706	5.67398	0.14984	2.84372	1.43900	4.68602	2.31770	3.74504	5.67782
1.51316	3.41618	0.11930	5.13320	4.51922	2.64836	1.66214	2.03138	2.21456	2.34980
0.68270	5.88158	0.15140	5.86022	5.14712	1.80008	2.27576	4.01990	1.32638	0.71672
2.55206	0.25148	0.14522	1.72466	3.14204	1.56752	4.66730	3.98846	5.46518	5.34134
5.48198	5.50676	2.52806	3.03446	2.32394	3.19352	5.81996	1.81130	4.42898	3.91400
5.94236	3.11930	4.65782	5.45900	6.02174	5.22014	5.03576	3.94484	4.07660	4.79720
2.36858	0.62888	3.81800	1.11050	3.76742	0.45182	0.84812	1.96520	2.34380	3.55958
2.33846	4.88660	5.28806	3.69542	1.81184	2.08820	1.97876	0.16796	2.54378	3.02720
3.69500	0.43304	3.77684	4.59662	0.29102	4.46696	2.29976	3.48026	1.20398	6.04850
5.09870	5.10056	0.45296	0.82772	0.26846	3.61586	5.77616	5.85794	5.43236	5.28770
4.08566	3.64766	5.09720	5.79566	1.38968	5.72594	4.97660	0.24872	2.40164	4.07234
4.73318	1.52918	3.45482	2.49140	2.61662	0.72896	3.73388	3.41090	4.09646	2.23220

Упорядоченная выборка:

X

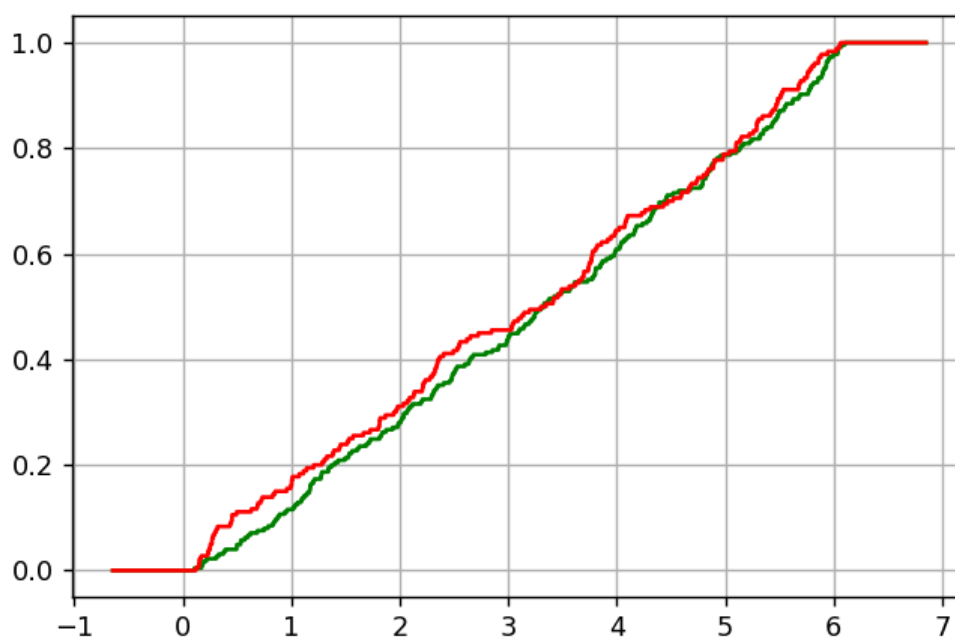
0.10372	0.16912	0.18010	0.18784	0.21754	0.30586	0.32992	0.37240	0.38986	0.49498
0.49612	0.52954	0.53296	0.57010	0.59668	0.62050	0.68506	0.74812	0.78484	0.82522
0.83896	0.85420	0.87382	0.88810	0.93814	0.95908	1.01638	1.03162	1.05436	1.08742
1.09852	1.11922	1.14088	1.16368	1.17004	1.17172	1.17760	1.19848	1.21030	1.27138
1.27390	1.27930	1.33936	1.34152	1.36768	1.39936	1.42936	1.49374	1.52398	1.53160
1.55722	1.60462	1.62460	1.68946	1.71376	1.72516	1.81198	1.82686	1.83604	1.87168
1.93174	1.98040	1.98514	2.00848	2.01586	2.03332	2.03500	2.06416	2.07454	2.09272
2.11474	2.19358	2.20612	2.28754	2.29270	2.30752	2.31490	2.33446	2.34778	2.40160
2.45788	2.47720	2.48062	2.48494	2.50318	2.51224	2.52532	2.59702	2.63464	2.63944
2.65498	2.67502	2.79004	2.85154	2.90710	2.91190	2.97988	2.98336	2.99590	2.99866
3.01468	3.09022	3.10180	3.13882	3.15160	3.18976	3.20950	3.22804	3.24352	3.24424
3.26290	3.29734	3.32386	3.32800	3.36118	3.37708	3.42166	3.45700	3.49174	3.56350
3.57148	3.60034	3.61018	3.72838	3.77314	3.78706	3.80044	3.80140	3.80344	3.84316
3.85360	3.86650	3.90286	3.94108	3.97648	3.98080	3.99022	4.02508	4.03246	4.03888
4.06408	4.08688	4.11088	4.15594	4.16296	4.17046	4.17952	4.23484	4.27534	4.30144
4.31086	4.31812	4.33204	4.33996	4.35748	4.38838	4.38868	4.44412	4.45528	4.46176
4.51744	4.57180	4.67602	4.77148	4.78906	4.79224	4.79428	4.81978	4.82668	4.83544
4.85422	4.86634	4.87192	4.88668	4.89724	4.92538	4.95424	5.04448	5.10202	5.12908
5.14078	5.16154	5.22262	5.24374	5.31676	5.32156	5.34892	5.35168	5.38360	5.42752
5.43328	5.44888	5.46724	5.48110	5.48848	5.49532	5.53840	5.54794	5.56102	5.61922
5.62390	5.67454	5.68678	5.76352	5.76826	5.78236	5.78992	5.81002	5.84986	5.85418
5.88634	5.89408	5.90020	5.90710	5.92648	5.93128	5.93236	5.94868	5.95954	5.99356
6.03304	6.03904	6.03988	6.06682	6.09958					

Y

0.11930	0.14522	0.14984	0.15140	0.16796	0.22784	0.23426	0.24872	0.25148	0.26750
0.26846	0.27560	0.29102	0.30458	0.31778	0.43304	0.44462	0.45182	0.45296	0.49358
0.62888	0.68270	0.68942	0.71672	0.72896	0.82772	0.84812	0.95696	0.99338	1.00220
1.00280	1.00880	1.07132	1.11050	1.13900	1.20398	1.28078	1.30232	1.32638	1.37864
1.38968	1.43900	1.45028	1.51316	1.52918	1.56752	1.66214	1.72466	1.80008	1.81130
1.81184	1.81586	1.86506	1.94120	1.96520	1.97876	2.03138	2.06816	2.08820	2.12486
2.13080	2.20448	2.21042	2.21456	2.23220	2.27576	2.29976	2.31770	2.32394	2.33846
2.34380	2.34980	2.36858	2.40164	2.49140	2.52806	2.54378	2.55206	2.61662	2.64836
2.72180	2.84372	3.02720	3.03446	3.04820	3.08882	3.11930	3.14204	3.19352	3.29726
3.34646	3.41090	3.41618	3.45482	3.48026	3.48920	3.55958	3.61586	3.64766	3.67544



3.69500	3.69542	3.73388	3.74090	3.74504	3.76202	3.76742	3.77204	3.77684	3.80342
3.81800	3.85706	3.91400	3.94484	3.97742	3.98846	4.01990	4.07234	4.07660	4.08566
4.09646	4.22588	4.25822	4.30694	4.42898	4.46696	4.51922	4.58864	4.59662	4.65782
4.66730	4.68602	4.73318	4.74554	4.79720	4.82846	4.84250	4.88660	4.89188	4.89818
4.97660	4.98080	5.03576	5.09720	5.09870	5.10056	5.13320	5.14712	5.22014	5.25854
5.28770	5.28806	5.29076	5.30564	5.34134	5.41478	5.43236	5.45900	5.46518	5.47946
5.48198	5.50676	5.51726	5.53082	5.67398	5.67782	5.69132	5.72594	5.75366	5.75636
5.77616	5.79566	5.81996	5.85794	5.86022	5.88158	5.94236	6.02174	6.04850	6.06098



$N$	$M$	$D_{N,M}$	$K_{N,M}$	$x^*$	$F_N(x^*)$	$F_N(x^* - 0)$	$F_M(x^*)$	$F_M(x^* - 0)$
225	180	0.071111	0.711111	0.49498	0.044444	0.040000	0.111111	0.111111

При втором способе проверки гипотезы о соответствии выборки равномерному распределению с помощью критерия Колмогорова значение функции:  $pval = 0.66656$

### Анализ результатов и выводы

Таблица критических значений:

$\alpha$	0.01	0.02	0.05	0.1	0.2
$K_\alpha$	1.63	1.57	1.36	1.22	1.07

Задание 1)

$D_N\sqrt{N}$	$\alpha$	$K_\alpha$	Верность гипотезы
0.91618	0.05	1.36	Да

$\alpha$	pval	Верность гипотезы
0.05000	0.35840	Да

Гипотеза о соответствии выборки равномерному распределению на отрезке  $[a, b]$  не противоречит экспериментальным данным (может быть принята) при уровне значимости  $\alpha = 0,05$ .

Задание 2)

$K_{N,M}$	$\alpha$	$K_\alpha$	Верность гипотезы
0.71111	0.05	1.36	Да

$\alpha$	pval	Верность гипотезы
0.05	0.66656	Да

Гипотеза об одинаковом распределении двух случайных выборок не противоречит экспериментальным данным (может быть принята) при уровне значимости  $\alpha = 0,05$ .

### **Список использованной литературы**

1. Лобузов А.А. Математическая статистика [Электронный ресурс]: Методические указания по выполнению лабораторных работ / под ред. Ю. И. Худака. Москва: Московский технологический университет (МИРЭА), 2017. 36 с.
2. Чернова Н. И. Математическая статистика: Учеб. пособие / Новосиб. гос. ун-т. Новосибирск, 2007. 148 с

### Приложение (Листинг программы)

```

import math
import numpy

import matplotlib.mlab as mlab
import matplotlib.pyplot as plt

import scipy.stats as pystats
import scipy

from docx import Document

kolm = {
    0.01 : 1.63,
    0.02 : 1.57,
    0.05 : 1.36,
    0.1 : 1.22,
    0.2 : 1.07
}

head_1_str = [
    'a', 'b', 'N', 'D_N', 'D_N sqrt{N}', 'x^*', 'F (x^*)', 'F_N (x^*)', 'F_N
(x^* - 0)'
]
head_2_str = [
    'N', 'M', 'D_N,M', 'K_N,M', 'x^*', 'F_N (x^*)', 'F_N (x^* - 0)', 'F_M
(x^*)', 'F_M (x^* - 0)'
]

head_3_str = [
    'D_N sqrt{N}', 'alpha', 'k_alpha', 'Верность гипотезы'
]
head_4_str = [
    'alpha', 'pval', 'Верность гипотезы'
]

head_5_str = [
    'K_N,M', 'alpha', 'k_alpha', 'Верность гипотезы'
]
head_6_str = [
    'alpha', 'pval', 'Верность гипотезы'
]

```

```
hyp_text = {
    True : 'Да',
    False : 'Нет'
}
```

```
def strm(a):
    n = len(a)
    b = a[0]
    m = len(b)
    for i in range(n):
        for k in range(m):
            if type(a[i][k]) == int:
                a[i][k] = str(a[i][k])
            elif type(a[i][k]) != str:
                a[i][k] = str('%0.5f' % a[i][k])
```

```
def strm_lite(a):
    n = len(a)
    b = a[0]
    m = len(b)
    for i in range(n):
        for k in range(m):
            if type(a[i][k]) == int:
                a[i][k] = str(a[i][k])
            elif type(a[i][k]) != str:
                a[i][k] = str(a[i][k])
```

```
def line_out(l, cols = 10):
    tbl = []
    q = []
    for i in range(len(l)):
        q.append(l[i])
        if (len(q) == 10) or (i+1 == len(l)):
            while(len(q) != 10):
                q.append("")
            tbl.append(q)
            q = []
    return tbl
```

```
def onetablein(docname):
    document = Document(docname)
    table = document.tables[0]
    array = []
```

```

for i in range(0,len(table.rows)):
    row = table.rows[i]
    for cell in row.cells:
        array.append(float(cell.text.replace(',','.')))
return array

```

```

ud2 = onetablein('UD-2_10.docx')
ud2a = 2.0
ud2b = 7.5
ud3 = onetablein('UD-3_10.docx')
print(ud2, ud2a, ud2b)
print()
print(ud3)

```

```

class UniSample():
    def __init__(self, array, a, b):
        self.array = [i for i in array]
        self.n = len(array)
        self.a = a
        self.b = b

    def Fj(self,x):
        s = 0
        for i in self.array:
            if i <= x:
                s = s + 1
        return s/self.n

    def Fjo(self,x):
        s = 0
        for i in self.array:
            if i < x:
                s = s + 1
        return s/self.n

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

    def Dn(self):

```

```

        a = [max([abs(self.Fj(i)-self.F(i)), abs(self.Fjo(i)-self.F(i))]) for i
in self.array]

```

```

        return(max(a))

```

```

    def xx(self):
        for i in self.array:
            if self.Dn() == max([abs(self.Fj(i)-self.F(i)), abs(self.Fjo(i)-
self.F(i))]):
                return i

```

```

    def out_1(self):
        x = self.xx()
        l = [
            self.a,
            self.b,
            self.n,
            self.Dn(),
            self.Dn()*(self.n**0.5),
            x,
            self.F(x),
            self.Fj(x),
            self.Fjo(x)
        ]
        return l

```

```

    def Graf(self, filename = 'Uni_Graf.png', dx = 0.00001):
        xmin = self.a - (self.b - self.a)/8
        xmax = self.b + (self.b - self.a)/8
        xlist = mlab.frange (xmin, xmax, dx)
        ylist = [self.F(x) for x in xlist]
        zlist = [self.Fj(x) for x in xlist]
        plt.plot(xlist, ylist, color = 'blue', alpha = 0.75, linewidth = 2)
        plt.plot(xlist, zlist, color = 'red', linewidth = 1.5)
        plt.grid(True)
        plt.savefig(filename, dpi = 128)
        plt.clf()
        print('Done! [gr_1]')

```

```

    def out_3(self, alpha = 0.05):
        dn = self.Dn()*(self.n**0.5)
        ka = kolm[alpha]
        l = [
            dn,
            str(alpha),
            str(ka),

```

```

        hyp_text[dn <= ka]
    ]
    return l

    def out_4(self, alpha = 0.05):
        g = pystats.kstest(self.array, scipy.stats.uniform(loc = self.a, scale
= self.b-self.a).cdf).pvalue
        l = [
            alpha,
            g,
            hyp_text[g >= alpha]
        ]
        return l

    def hyp(self, alpha = 0.05):
        dn = self.Dn()*(self.n**0.5)
        ka = kolm[alpha]
        return dn <= ka

us = UniSample(ud2, ud2a, ud2b)
print(us.out_1())
us.Graf(dx = 0.01)
print(us.out_3())
print(us.out_4())

class DoubleSample():
    def __init__(self, arr_1, arr_2):
        self.arr_1 = [i for i in arr_1]
        self.arr_2 = [i for i in arr_2]
        self.n = len(arr_1)
        self.m = len(arr_2)
        self.a = min(sorted(arr_1)[0], sorted(arr_2)[0])
        self.b = max(sorted(arr_1)[self.n - 1], sorted(arr_2)[self.m - 1])

    def Fn(self,x):
        s = 0
        for i in self.arr_1:
            if i <= x:
                s = s + 1
        return s/self.n

    def Fno(self,x):
        s = 0
        for i in self.arr_1:

```



```

        if i < x:
            s = s + 1
        return s/self.n

def Fm(self,x):
    s = 0
    for i in self.arr_2:
        if i <= x:
            s = s + 1
    return s/self.m

def Fmo(self,x):
    s = 0
    for i in self.arr_2:
        if i < x:
            s = s + 1
    return s/self.m

def Dnm(self):
    l = []
    l.extend([abs(self.Fn(i) - self.Fm(i)) for i in self.arr_1])
    l.extend([abs(self.Fno(i) - self.Fm(i)) for i in self.arr_1])
    l.extend([abs(self.Fn(i) - self.Fm(i)) for i in self.arr_2])
    l.extend([abs(self.Fn(i) - self.Fmo(i)) for i in self.arr_2])
    return max(l)

def Knm(self):
    return ((self.n*self.m/(self.n+self.m))**0.5)*self.Dnm()

def xx(self):
    for i in self.arr_1:
        if (self.Dnm() == abs(self.Fn(i) - self.Fm(i))):
            return i
        if (self.Dnm() == abs(self.Fno(i) - self.Fm(i))):
            return i
    for i in self.arr_1:
        if (self.Dnm() == abs(self.Fn(i) - self.Fm(i))):
            return i
        if (self.Dnm() == abs(self.Fn(i) - self.Fmo(i))):
            return i

def out_2(self):
    x = self.xx()
    l = [
        self.n,
        self.m,

```

```

        self.Dnm(),
        self.Knm(),
        x,
        self.Fn(x),
        self.Fno(x),
        self.Fm(x),
        self.Fmo(x)
    ]
    return l

def Graf(self, filename = 'Double_Graf.png', dx = 0.00001):
    xmin = self.a - (self.b - self.a)/8
    xmax = self.b + (self.b - self.a)/8
    xlist = mlab.frange (xmin, xmax, dx)
    ylist = [self.Fn(x) for x in xlist]
    zlist = [self.Fm(x) for x in xlist]
    plt.plot(xlist, ylist, color = 'Green', linewidth = 1.5)
    plt.plot(xlist, zlist, color = 'red', linewidth = 1.5)
    plt.grid(True)
    plt.savefig(filename, dpi = 128)
    plt.clf()
    print('Done! [gr_2]')

def out_5(self, alpha = 0.05):
    Knm = self.Knm()
    ka = kolm[alpha]
    l = [
        Knm,
        str(alpha),
        str(ka),
        hyp_text[Knm <= ka]
    ]
    return l

def out_6(self, alpha = 0.05):
    g = scipy.stats.ks_2samp(self.arr_1, self.arr_2).pvalue
    l = [
        str(alpha),
        g,
        hyp_text[g >= alpha]
    ]
    return l

def hyp(self, alpha = 0.05):
    Knm = self.Knm()
    ka = kolm[alpha]

```

```
return Knm <= ka
```

```
usd = DoubleSample(ud2, ud3)
print()
print(head_2_str)
print(usd.out_2())
print()
#usd.Graf(dx = 0.01)
print(head_5_str)
print(usd.out_5())
print()
print(head_6_str)
print(usd.out_6())
```

```
def tabler(document, out, head = None):
```

```
    tl = []
    if head != None:
        tl.append(head)
    for i in out:
        tl.append(i)
    strm(tl)
```

```
    table = document.add_table(rows = len(tl), cols = len(tl[0]))
    for i in range(len(tl)):
        hdr_cells = table.rows[i].cells
        for k in range(len(tl[0])):
            if type(tl[i][k]) == str:
                hdr_cells[k].text = tl[i][k]
            else:
                pass #LaTeh
```

```
text_anal_1 = {
    True : 'Гипотеза о соответствии выборки равномерному
распределению на отрезке [a, b] не противоречит экспериментальным
данным (может быть принята) при уровне значимости alpha = 0,05.',
    False : 'Гипотеза о соответствии выборки равномерному
распределению на отрезке [a, b] противоречит экспериментальным
данным (не может быть принята) при уровне значимости alpha = 0,05.'
}
```

```
text_anal_2 = {
```

True : 'Гипотеза об одинаковом распределении двух случайных выборок не противоречит экспериментальным данным (может быть принята) при уровне значимости  $\alpha = 0,05$ .'

True : 'Гипотеза об одинаковом распределении двух случайных выборок противоречит экспериментальным данным (не может быть принята) при уровне значимости  $\alpha = 0,05$ .'

}

```
def doccreator(arr_1, arr_2, a,b, filename = 'some'):
    document = Document()
    document.add_paragraph('Результаты расчетов')

    document.add_paragraph('Задание 1')
    tbl = line_out(arr_1)
    document.add_paragraph('Полученная выборка:')
    tabler(document,tbl)
    document.add_paragraph("")
    tbl = line_out(sorted(arr_1))
    document.add_paragraph('Упорядоченная выборка:')
    tabler(document, tbl)
    document.add_paragraph("")
    us = UniSample(arr_1, a, b)
    us.Graf(filename = filename + '_1.png')
    document.add_picture(filename + '_1.png')
    document.add_paragraph("")
    out_1 = us.out_1()
    tabler(document, [out_1], head_1_str)
    document.add_paragraph("")
    s = 'При втором способе проверки гипотезы о соответствии'
    s = s + ' выборки равномерному распределению с помощью'
    s = s + ' критерия Колмогорова значение функции: pval = '
    s = s + str("%.5f" % us.out_4()[1])
    document.add_paragraph(s)

    document.add_paragraph("")
    document.add_paragraph("")

    document.add_paragraph('Задание 2')
    document.add_paragraph('Полученная выборка:')
    document.add_paragraph('X')
    tbl = line_out(arr_1)
    tabler(document,tbl)
    document.add_paragraph("")
```

```

document.add_paragraph('Y')
tbl = line_out(arr_2)
tabler(document,tbl)
document.add_paragraph("")
document.add_paragraph('Упорядоченная выборка:')
document.add_paragraph('X')
tbl = line_out(sorted(arr_1))
tabler(document,tbl)
document.add_paragraph("")
document.add_paragraph('Y')
tbl = line_out(sorted(arr_2))
tabler(document,tbl)
document.add_paragraph("")
ds = DoubleSample(arr_1, arr_2)
ds.Graf(filename = filename + '_2.png')
document.add_picture(filename + '_2.png')
out_2 = ds.out_2()
tabler(document, [out_2], head_2_str)
s = 'При втором способе проверки гипотезы о соответствии'
s = s + ' выборки равномерному распределению с помощью'
критерия Колмогорова значение функции: pval = '
s = s + str("%.5f"% ds.out_6()[1])
document.add_paragraph(s)

document.add_paragraph("")
document.add_paragraph("")

document.add_paragraph('Анализ результатов и выводы')
document.add_paragraph('Таблица критических значений:')
tbl = [['alpha'], ['k_alpha']]
tbl[0].extend([str(i) for i in kolm])
tbl[1].extend([str(kolm[i]) for i in kolm])
tabler(document, tbl)
document.add_paragraph("")
document.add_paragraph('Задание 1')
tabler(document, [us.out_3()], head_3_str)
document.add_paragraph("")
tabler(document, [us.out_4()], head_4_str)
document.add_paragraph(text_anal_1[us.hyp()])
document.add_paragraph("")
document.add_paragraph('Задание 2')
tabler(document, [ds.out_5()], head_5_str)
document.add_paragraph("")
tabler(document, [ds.out_6()], head_6_str)
document.add_paragraph(text_anal_2[ds.hyp()])
document.add_paragraph("")

```

```
document.save(filename + '.docx')  
print('Done! [doc]')
```

```
doccreator(ud2, ud3, ud2a, ud2b)
```