Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Ульяновский государственный технический университет»

Лабораторная работа №1

Дисциплина: «Методы моделирования»

Тема: «Анализ и генерация случайных чисел. Основы имитационного моделирования»

Вариант 2

Выполнил

Студент группы УПАСбд-31

Джураев И.Д.

Проверил

преподаватель кафедры

«Вычислительная техника»

Валюх В. В.

Ульяновск, 2024

**Цель работы:** Изучение основных характеристик случайных величин на базе теории вероятностей и математической статистики; изучение и программирование способов получения псевдослучайных чисел.

***ЗАДАНИЕ 1***

Для стандартного генератора случайных чисел выбранного Вами языка программирования получить три последовательности  N случайных чисел (N={100, 1000, 10000}), для которых определить следующие характеристики: математическое ожидание М, дисперсию D и среднеквадратичное отклонение .

Выполнить проверку частотности и равномерности генератора.

Построить графики  Функций Р(X) для оценки частотности генератора. Для получаемой выборки  N  чисел  Р(X) – вероятность попадания генерируемой случайной величины в соответствующий интервал ее области определения.

Сравнить результаты с теоретическими.

Для оценки равномерности генератора случайных чисел выполнить расчет математического ожидания Mi для i  последовательностей  из 1000 случайных чисел (i=1,2,..., 10) и для i последовательностей случайных чисел переменной длины (длина  i-ой  последовательности задается как i\*1000;  i = 1, 2, ..., 10). Построить графики зависимости разности (М-Мi) от номера  последовательности i, где  М - теоретическое  математическое ожидание равномерного распределения случайных чисел,  Мi - расчетное математическое ожидание для  i-й последовательности случайных чисел,  полученных от генератора. По данным результатам определить  Р{|М-Мi|<s} - вероятность того, что отклонения расчетного математического ожидания от теоретического не превышают величину теоретического среднеквадратичного отклонения.

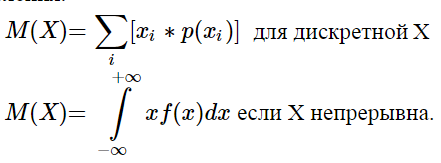
***ЗАДАНИЕ 2***

Запрограммировать заданный вариант генератора случайных чисел и выполнить для него задание 1.

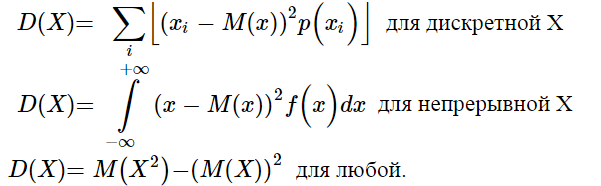
**Описание реализации и теория**

Для начала выполнения лабораторной работы начнем с реализации стандартного генератора случайных чисел на языке питон и расчета для полученных трех последовательностей (N={100, 1000, 10000}) таких характеристик, как математическое ожидание (M), дисперсия (D) и среднеквадратичное отклонение (σ). Нам понадобится использовать следующие библиотеки: random для генерации случайных чисел, numpy для расчета статистики и matplotlib для визуализации.

**Математическое ожидание**взвешенная по вероятности средняя величина всех возможных значений X, определяющая меру центральности распределения.

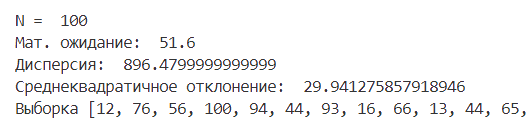


**Дисперсией**случайной переменной называется  второй  момент среднего, являющийся мерой разброса вероятностного распределения.

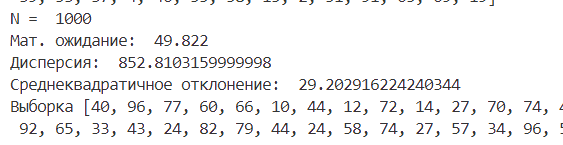


**Среднеквадратичным отклонением** случайной величины называется квадратный корень из дисперсии этой величины: σ=√D(X).

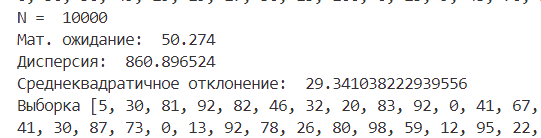
**Задание 1**



Для N 100



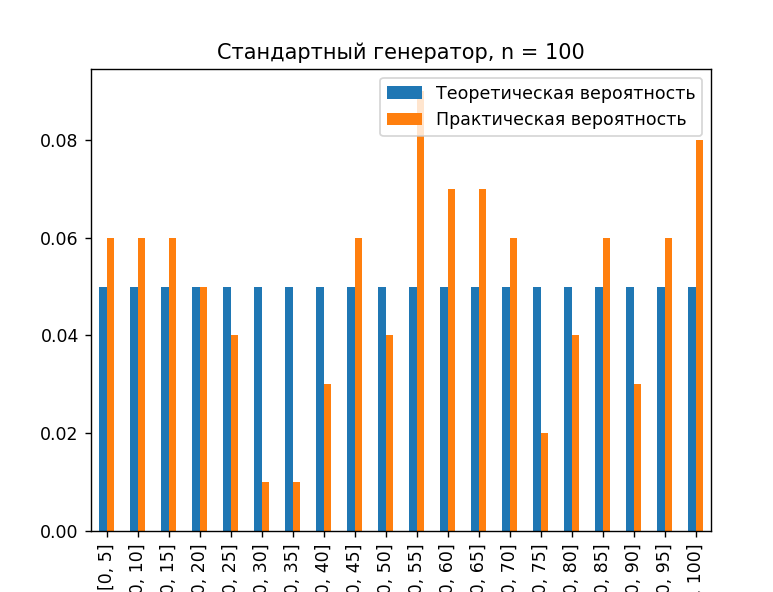
Для N 1000



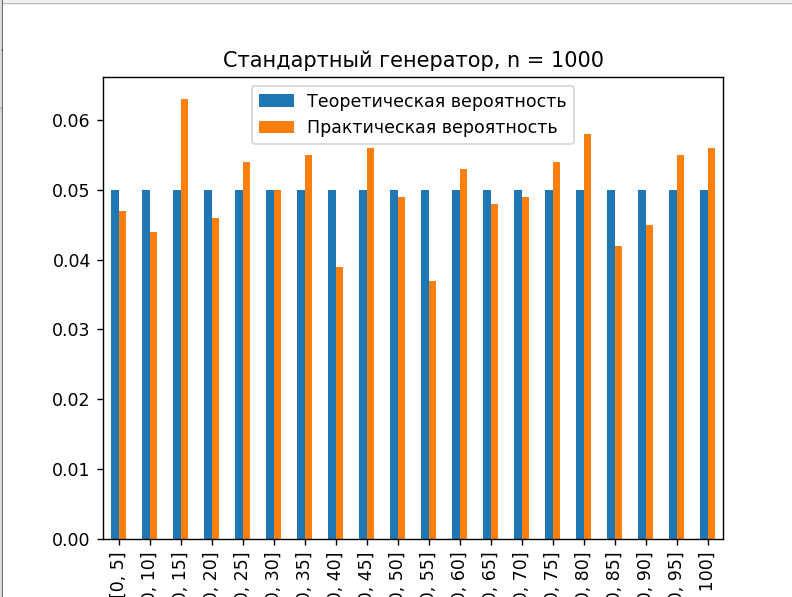
Для N 10000

Для проверки частотности генерируемых случайных чисел мы разделим выборку N на 20 равных интервалов и построим графики функции P(X) для трех различных выборок.

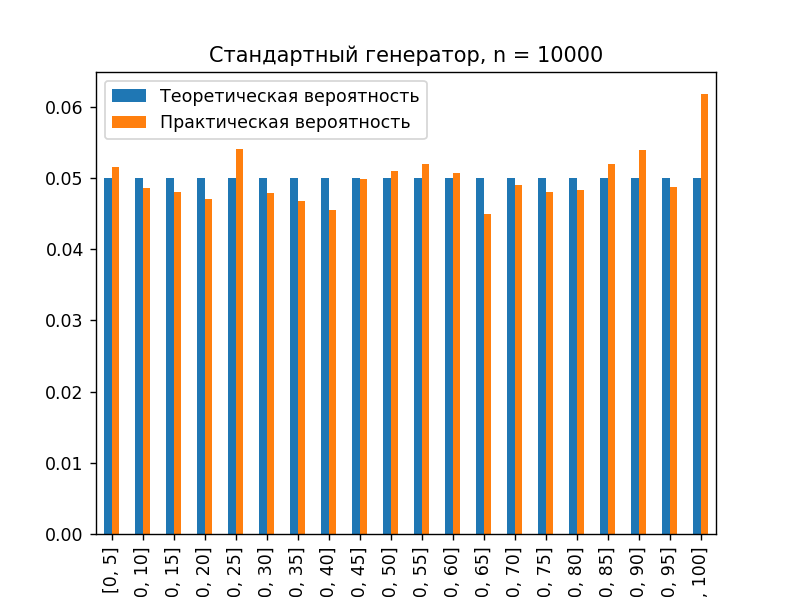
Теоретическая вероятность, что число попадет в один из этих интервалов, составляет p(x) = P (вероятность) / количество интервалов =



Для N 100



Для N 1000



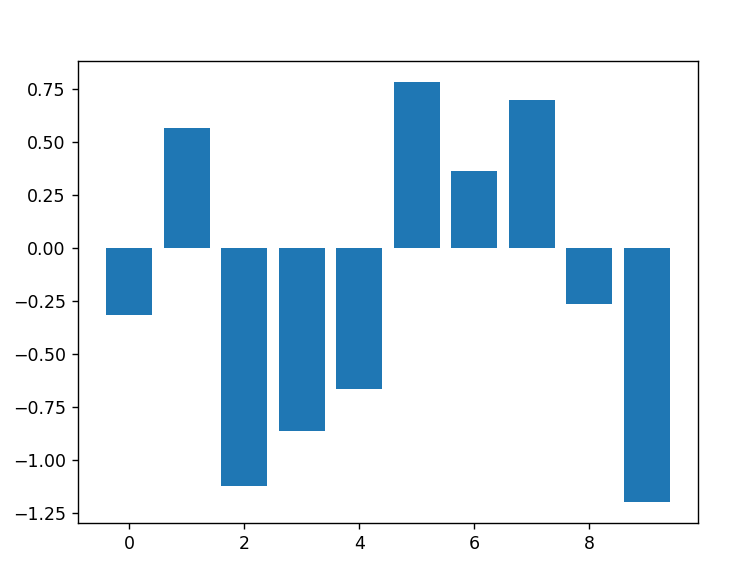
Для N 10000

На графиках видно, что при уменьшении размера выборки практическая вероятность начинает сильнее отличаться от теоретической вероятности.

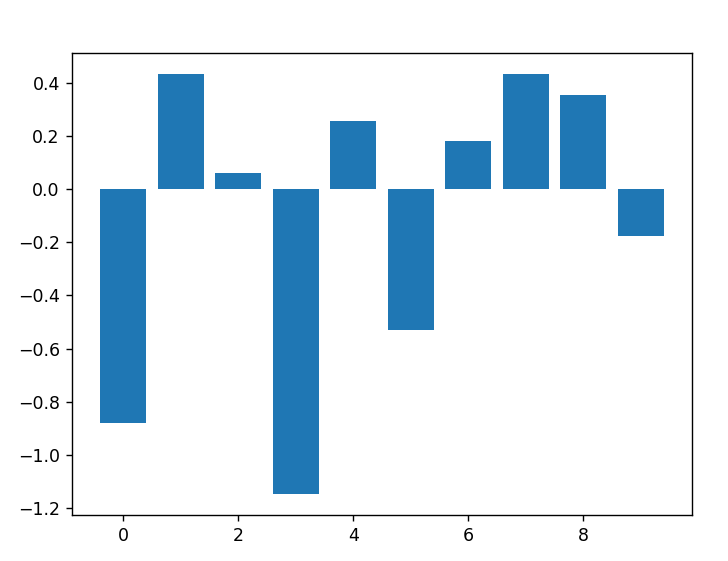
Чтобы оценить равномерность генератора случайных чисел, мы вычислим математическое ожидание Mi для i последовательностей, каждая из которых состоит из 1000 случайных чисел, а также для i последовательностей случайных чисел с переменными размерами выборок.

Теоретическое математическое ожидание M можно расчитать как среднее значение между a и b, то есть M = (a + b) / 2 = (0 + 100) / 2 = 50.

Затем мы построим графики, которые показывают зависимость разности (M - Mi) от номера последовательности i.



Зависимость разности (M - Mi) i,n=1000

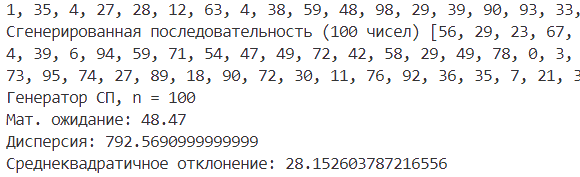


Зависимость разности (M - Mi) i,n=1000\*i

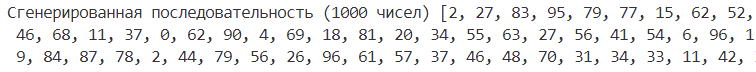
Теоретическое среднеквадратичное отклонение δ= 29.495

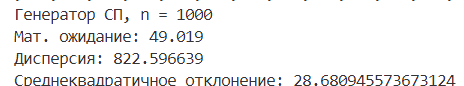
Как видно из графиков, отклонения расчетного математического ожидания от теоретического не превышают величину теоретического среднеквадратичного отклонения, и поэтому вероятность этого события равна 100%.

**Задание 2**

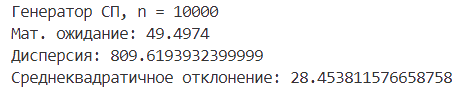


N=100



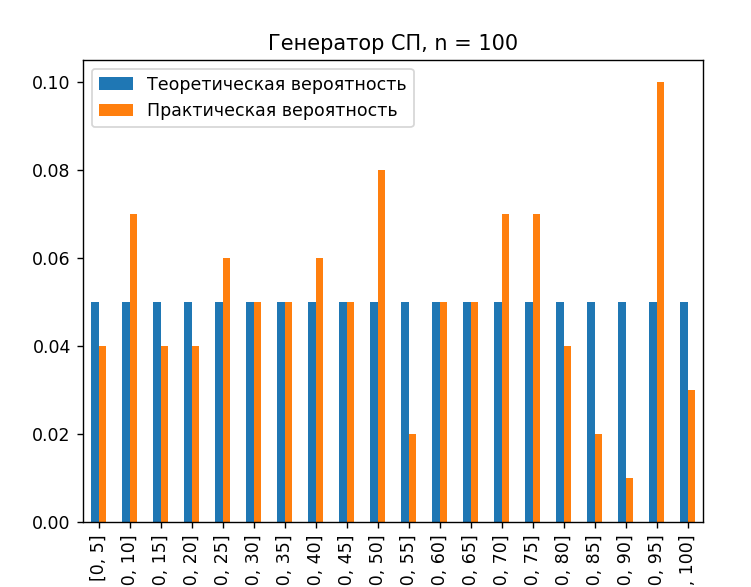


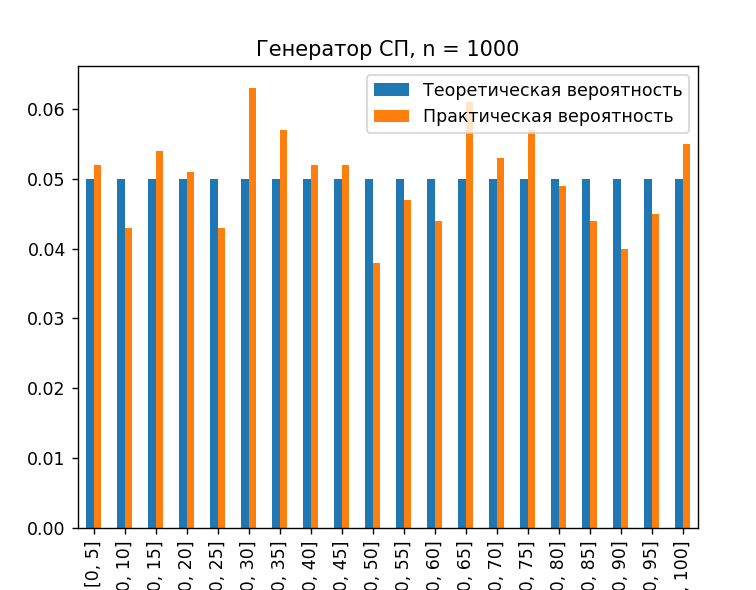
N=1000

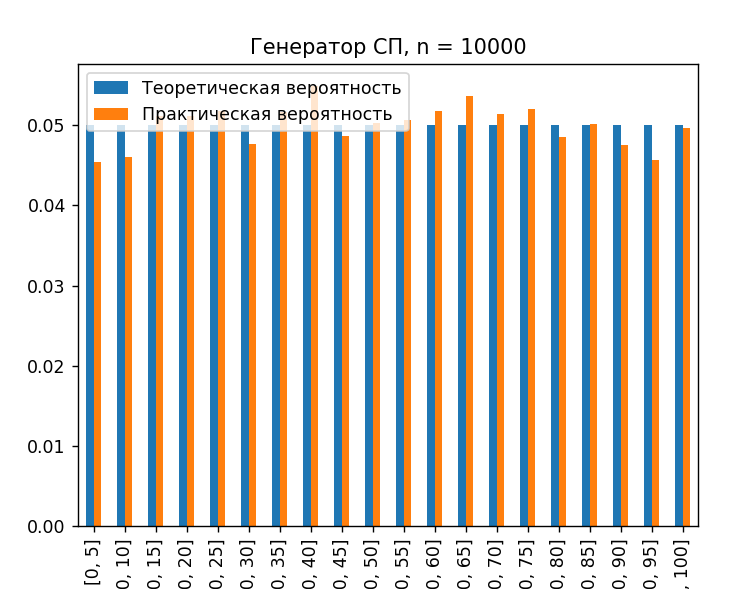


N=10000

Построим графики функций P(X)

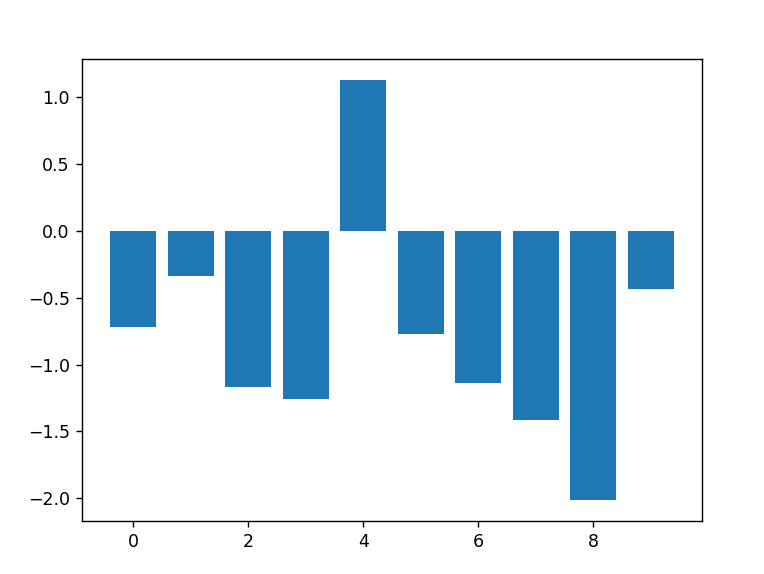


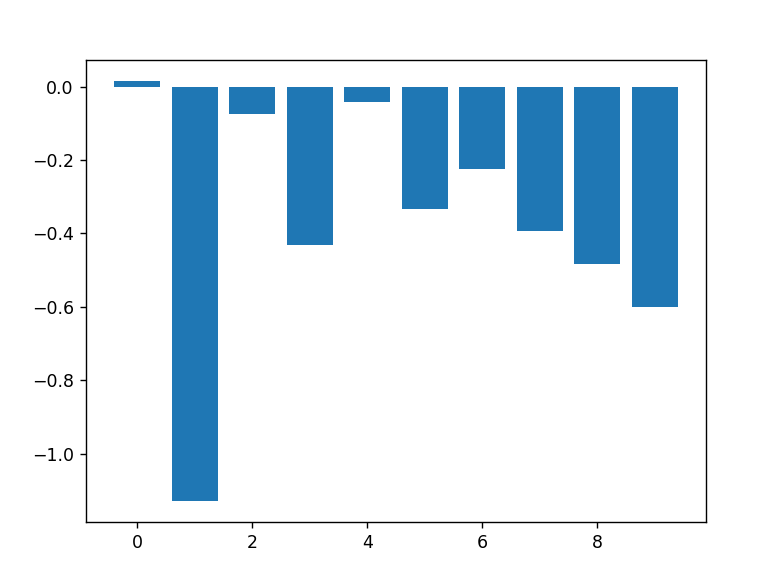




На графиках видно, что при уменьшении размера выборки практическая вероятность начинает сильнее отличаться от теоретической вероятности.

Построим графики, которые показывают зависимость разности (M - Mi) от номера последовательности i.





Теоретическое среднеквадратичное отклонение δ=28.429

Как видно из графиков, отклонения расчетного математического ожидания от теоретического не превышают величину теоретического среднеквадратичного отклонения, и поэтому вероятность этого события равна 100%.

|  |  |  |
| --- | --- | --- |
| Для 100 | | Для 100 |
| Мат ожидание | 51.6 | 48.47 |
| Дисперсия | 896.47 | 792.56 |
| Ср.кв.отклонение | 29.94 | 28.15 |
| Для 1000 | | Для 1000 |
| Мат ожидание | 49.82 | 49.01 |
| Дисперсия | 852.84 | 822.59 |
| Ср.кв.отклонение | 29.20 | 28.68 |
| Для 10000 | | Для 10000 |
| Мат ожидание | 50.27 | 49.49 |
| Дисперсия | 860.89 | 809.61 |
| Ср.кв.отклонение | 29.0666 | 28.45 |

**Выводы о проделанной работе**

Выполняя данную лабораторную работу, были изучены основные характеристики случайных величин на базе теории вероятностей и математической статистики; изучено программирование способов получения псевдослучайных чисел.

import random as rn

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

mass\_random = []

N = [100, 1000, 10000]

for i in N:

mass\_tmp = []

for j in range(i):

mass\_tmp.append(rn.randint(0, 100))

mass\_random.append(mass\_tmp)

M\_mass\_random = []

for i in range(3):

M\_mass\_random.append(np.mean(mass\_random[i]))

D\_mass\_random = []

for i in range(3):

D\_mass\_random.append(np.var(mass\_random[i]))

STD\_mass\_random = [np.std(mass\_random[0]),

np.std(mass\_random[1]),

np.std(mass\_random[2])]

for i in range(3):

print('N = ', N[i])

print('Мат. ожидание: ', M\_mass\_random[i])

print('Дисперсия: ', D\_mass\_random[i])

print('Среднеквадратичное отклонение: ', STD\_mass\_random[i])

print(f'Выборка {mass\_random[i]}')

def kol(x):

kol = []

for i in range(0, 100, 5):

kol.append(x.count(i)+

x.count(i + 1)+

x.count(i + 2)+

x.count(i + 3)+

x.count(i + 4))

kol[-1] += x.count(100)

return kol

G = [kol(mass\_random[i]) for i in range(3)]

G\_p = []

for i in range(3):

G\_p.append([G[i][j]/N[i] for j in range(len(G[i]))])

x1 = [1/20 for i in range(20)]

index = np.arange(20)

for i in range(3):

data = {'Теоретическая вероятность': x1,

'Практическая вероятность': G\_p[i]}

df = pd.DataFrame(data)

df.plot(kind='bar', title='Стандартный генератор, n = ' + str(N[i]))

plt.xticks(index,['[0, ' + str(i) + ']' for i in range(5, 101, 5)])

M\_1 = []

for i in range(10):

tmp = []

for i in range(1000):

tmp.append(rn.randint(0, 100))

M\_1.append(tmp)

plt.figure('Зависимость разности (M - Mi) i,n=1000')

plt.bar(list(range(10)), [np.mean(M\_1[i]) - 50 for i in range(10)])

M\_2 = []

ii = list(range(1, 11))

for i in range(10):

tmp = []

for i in range(ii[i] \* 1000):

tmp.append(rn.randint(0, 100))

M\_2.append(tmp)

plt.figure('Зависимость разности (M - Mi) i,n=i\*1000')

plt.bar(list(range(10)), [np.mean(M\_2[i]) - 50 for i in range(10)])

def random1():

seed1 = rn.randint(10, 99)

seed2 = rn.randint(10, 99)

s = str(seed1 \* seed2)

while len(s) != 4:

s = "0" + s

return int(s[1:3])

mass\_random100 = []

for j in range(100):

mass\_random100.append(random1())

print(f'Сгенерированная последовательность (100 чисел) {mass\_random100}')

print(f'Генератор СП, n = 100')

print(f'Мат. ожидание: {np.mean(mass\_random100)}')

print(f'Дисперсия: {np.var(mass\_random100)}')

print(f'Среднеквадратичное отклонение: {np.std(mass\_random100)}')

mass\_random1000 = []

for j in range(1000):

mass\_random1000.append(random1())

print(f'Сгенерированная последовательность (1000 чисел) {mass\_random1000}')

print(f'Генератор СП, n = 1000')

print(f'Мат. ожидание: {np.mean(mass\_random1000)}')

print(f'Дисперсия: {np.var(mass\_random1000)}')

print(f'Среднеквадратичное отклонение: {np.std(mass\_random1000)}')

mass\_random10000 = []

for j in range(10000):

mass\_random10000.append(random1())

print(f'Сгенерированная последовательность (10000 чисел) {mass\_random10000}')

print(f'Генератор СП, n = 10000')

print(f'Мат. ожидание: {np.mean(mass\_random10000)}')

print(f'Дисперсия: {np.var(mass\_random10000)}')

print(f'Среднеквадратичное отклонение: {np.std(mass\_random10000)}')

G\_1 = kol(mass\_random100)

G\_p\_1 = [G\_1[j]/100 for j in range(len(G\_1))]

x1 = [1/20 for i in range(20)]

index = np.arange(20)

data = {'Теоретическая вероятность': x1,

'Практическая вероятность': G\_p\_1}

df = pd.DataFrame(data)

df.plot(kind='bar', title='Генератор СП, n = 100')

plt.xticks(index,['[0, ' + str(i) + ']' for i in range(5, 101, 5)])

G\_2 = kol(mass\_random1000)

G\_p\_2 = [G\_2[j]/1000 for j in range(len(G\_2))]

x1 = [1/20 for i in range(20)]

index = np.arange(20)

data = {'Теоретическая вероятность': x1,

'Практическая вероятность': G\_p\_2}

df = pd.DataFrame(data)

df.plot(kind='bar', title='Генератор СП, n = 1000')

plt.xticks(index,['[0, ' + str(i) + ']' for i in range(5, 101, 5)])

G\_3 = kol(mass\_random10000)

G\_p\_3 = [G\_3[j]/10000 for j in range(len(G\_3))]

x1 = [1/20 for i in range(20)]

index = np.arange(20)

data = {'Теоретическая вероятность': x1,

'Практическая вероятность': G\_p\_3}

df = pd.DataFrame(data)

df.plot(kind='bar', title='Генератор СП, n = 10000')

plt.xticks(index,['[0, ' + str(i) + ']' for i in range(5, 101, 5)])

M\_1\_1000 = []

for i in range(10):

tmp = []

for i in range(1000):

tmp.append(random1())

M\_1\_1000.append(tmp)

plt.figure('Зависимость разности (M - Mi) i,n=1000. Генератор СП')

plt.bar(list(range(10)), [np.mean(M\_1\_1000[i]) - 50 for i in range(10)])

M\_2\_1000\_i = []

for i in range(10):

tmp = []

for i in range(ii[i] \* 1000):

tmp.append(random1())

M\_2\_1000\_i.append(tmp)

plt.figure('Зависимость разности (M - Mi) i,n=i\*1000. Генератор СП')

plt.bar(list(range(10)), [np.mean(M\_2\_1000\_i[i]) - 50 for i in range(10)])

plt.show()