##### Санкт-Петербургский Политехнический Университет

##### Петра Великого

##### Институт компьютерных наук и технологий

##### Высшая школа программной инженерии

Изображение выглядит как текст

Автоматически созданное описание

ЛАБОРАТОРНАЯ РАБОТА № 3

**Способы получения случайных чисел с требуемым законом распределения законом**

|  |  |  |  |
| --- | --- | --- | --- |
| Выполнила | |  |  |
| студентка гр. 51300202/00201 | |  | Козлова Е.А. |
|  | |  |  |
|  |
| Проверил | |  | Чуркин В.В |
|  | |  |  |

Санкт-Петербург  
2023 г.

Оглавление

[Цель работы 3](#_Toc128750880)

[Равномерное распределение 4](#_Toc128750881)

[Нормальное распределение 5](#_Toc128750882)

[Экспоненциальное распределение 7](#_Toc128750883)

[Хи-квадрат распределение 8](#_Toc128750884)

[Распределение Стьюдента 10](#_Toc128750885)

# Цель работы

1.Практическое освоение методов получения случайных величин, имеющих непрерывный характер распределения.

2.Разработка программных датчиков дискретных случайных величин.

3.Оценка точности моделирования: вычисление математического ожидания и дисперсии, сравнение полученных оценок с соответствующими теоретическими значениями.

4.Гpафическое пpедставление плотности распределения и интегральной функции pаспpеделения.

# Равномерное распределение

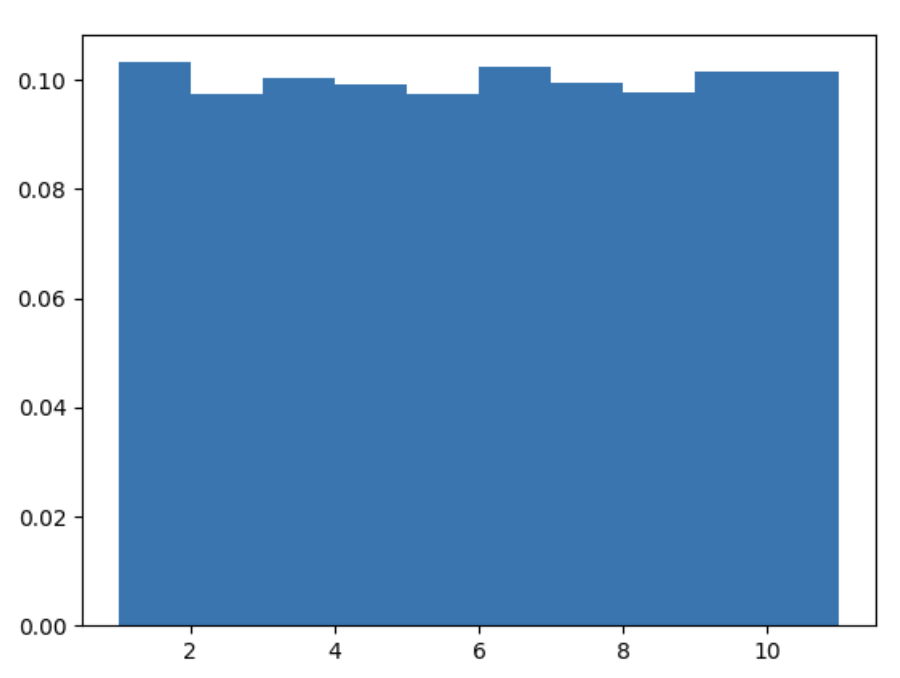
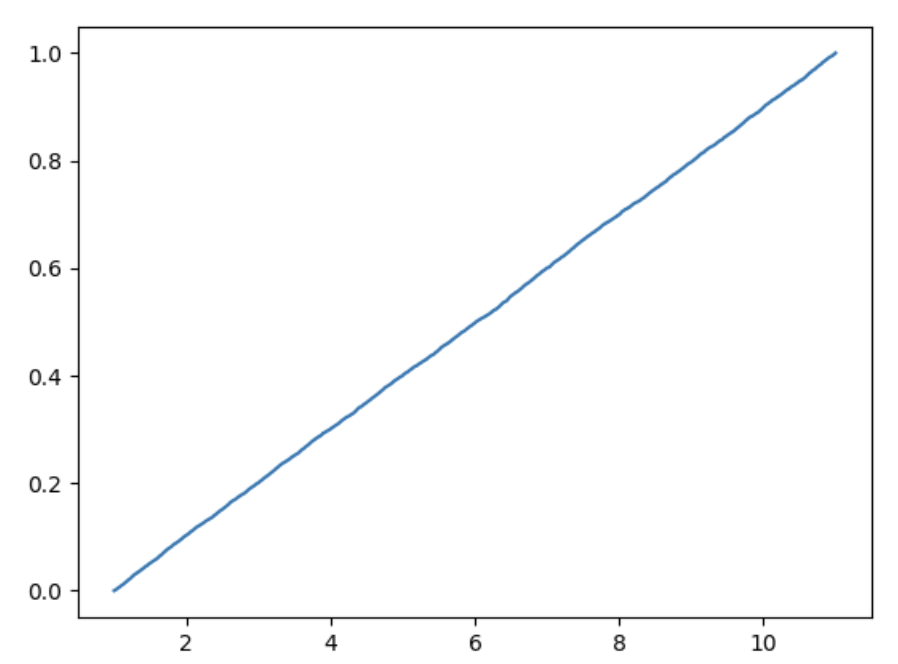
Код

**import** numpy **as** np  
**from** matplotlib **import** pyplot **as** plt  
**from** scipy.stats **import** norm  
**import** math  
  
**def** P(a, b):  
 u = np.random.uniform(0, 1, 1)  
 **return** (b-a)\*u + a  
  
n = 10000  
r = []  
**for** i **in** range(1, n):  
 r.append(P(1, 11)[0])  
print(r)  
M = sum(r) / n  
print(**"M: "**, M)  
print(**"D: "**, sum(R \*\* 2 **for** R **in** r) / n - M \*\* 2)  
  
**def** picture(r, n):  
 *# 1 график* plt.hist(r, density=**True**, label=**"Histogram"**, fill=**True**)  
 plt.show()  
  
 *# 2 график* tmp = [i/n **for** i **in** range(1, n)]  
 plt.plot(sorted(r), tmp)  
 plt.show()  
picture(r, n)

Результаты



Графики

# Нормальное распределение

Код

**import** numpy **as** np  
**from** matplotlib **import** pyplot **as** plt  
**from** scipy.stats **import** norm  
**import** math

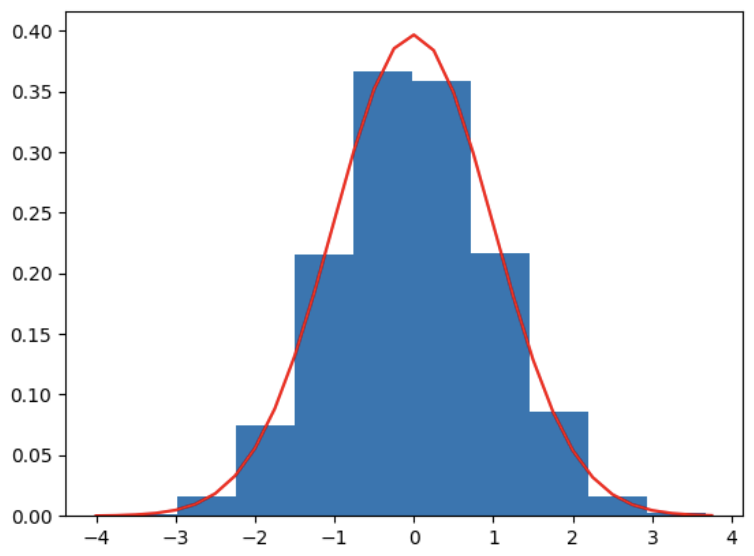
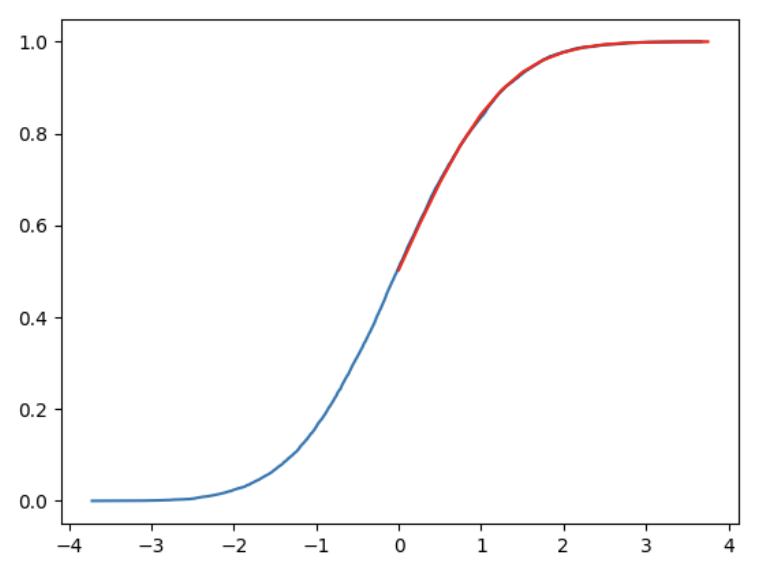
**def** picture(r, n):  
 *# 1 график* plt.hist(r, density=**True**, label=**"Histogram"**)  
 x = np.arange(-4, 4, 0.25)  
 y = []  
 **for** i **in** range(len(x)):  
 y.append(norm.pdf(x[i], np.mean(r), np.std(r)))  
  
 plt.plot(x, y, color=**'r'**)  
 plt.show()  
  
 *# 2 график* tmp = [i/n **for** i **in** range(1, n)]  
 plt.plot(sorted(r), tmp)  
  
 x = np.arange(0, 4, 0.25)  
 y = norm.cdf(x, np.mean(r), np.std(r))  
 plt.plot(x, y, color=**'r'**)  
 plt.show()  
  
**def** alg\_1():  
 a = np.random.uniform(0, 1, 12)  
 **return** sum(a) - 6  
**def** alg\_2():  
 a1 = np.random.uniform(0, 1, 1)  
 a2 = np.random.uniform(0, 1, 1)  
 **return** math.sqrt(-2 \* math.log(a2)) \* math.cos(2 \* math.pi \* a1)  
  
n = 10000  
r1 = []  
r2 = []  
**for** i **in** range(1, n):  
 r1.append(alg\_1())  
 r2.append(alg\_1())  
  
M = sum(r1) / n  
print(**"M: "**, M)  
print(**"D: "**, sum(R \*\* 2 **for** R **in** r1) / n - M \*\* 2)  
  
M = sum(r2) / n  
print(**"M: "**, M)  
print(**"D: "**, sum(R \*\* 2 **for** R **in** r2) / n - M \*\* 2)  
  
picture(r1, n)  
picture(r2, n)

Результаты

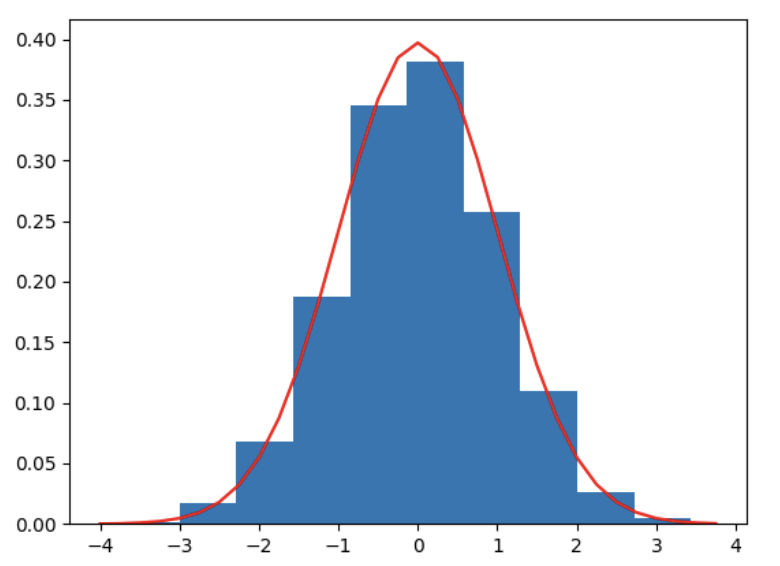
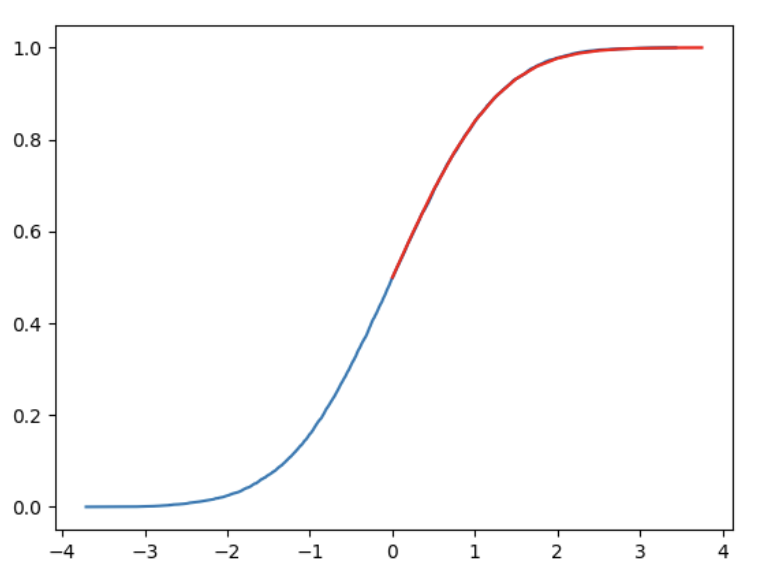


Графики

*Алгоритм №1*

*Алгоритм №2*

** **

# Экспоненциальное распределение

Код

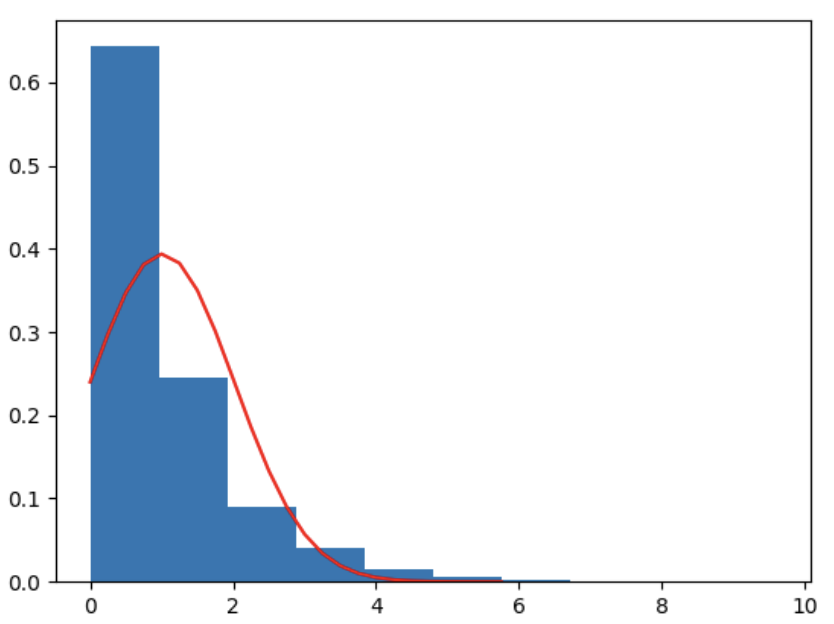
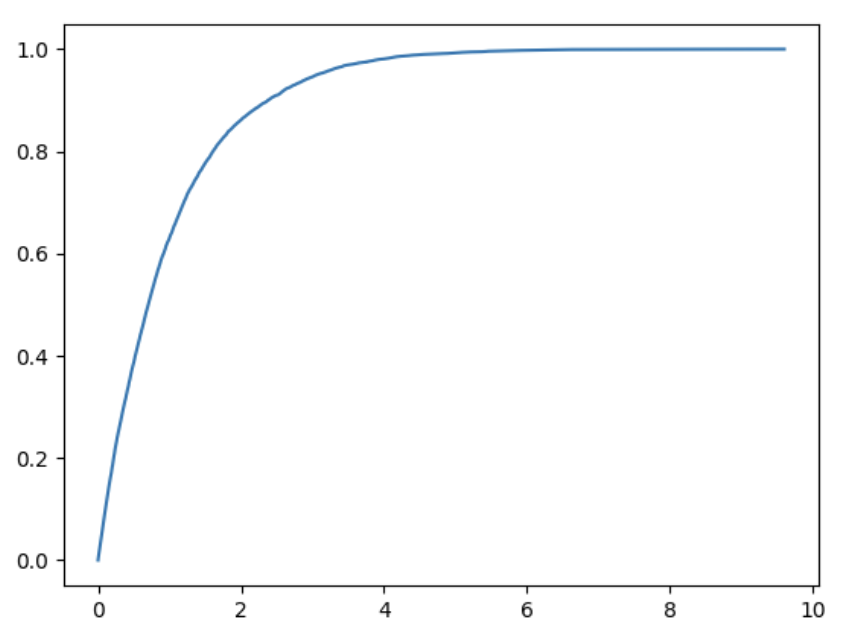
**import** numpy **as** np  
**from** matplotlib **import** pyplot **as** plt  
**from** scipy.stats **import** norm  
**import** math

**def** picture(r, n):  
 *# 1 график* plt.hist(r, density=**True**, label=**"Histogram"**)  
 x = np.arange(0, 6, 0.25)  
 y = []  
 **for** i **in** range(len(x)):  
 y.append(norm.pdf(x[i], np.mean(r), np.std(r)))  
  
 plt.plot(x, y, color=**'r'**)  
 plt.show()  
  
 *# 2 график* tmp = [i/n **for** i **in** range(1, n)]  
 plt.plot(sorted(r), tmp)  
 plt.show()  
  
**def** P(b):  
 a = np.random.uniform(0, 1, 1)  
 **return** -b \* math.log(a)  
  
n = 10000  
r = []  
b = 1  
**for** i **in** range(1, n):  
 r.append(P(b))  
  
M = sum(r) / n  
print(**"M: "**, M)  
print(**"D: "**, sum(R \*\* 2 **for** R **in** r) / n - M \*\* 2)  
  
picture(r, n)

Результаты



Графики

# Хи-квадрат распределение

Код

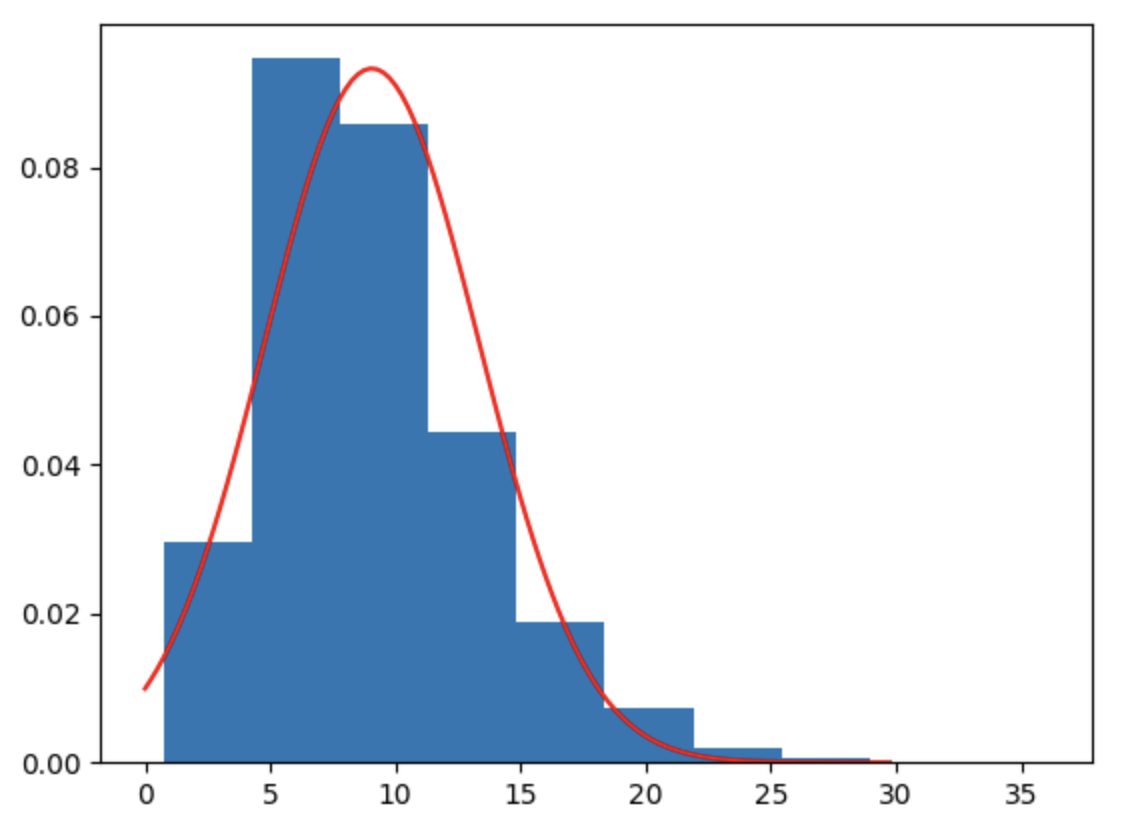
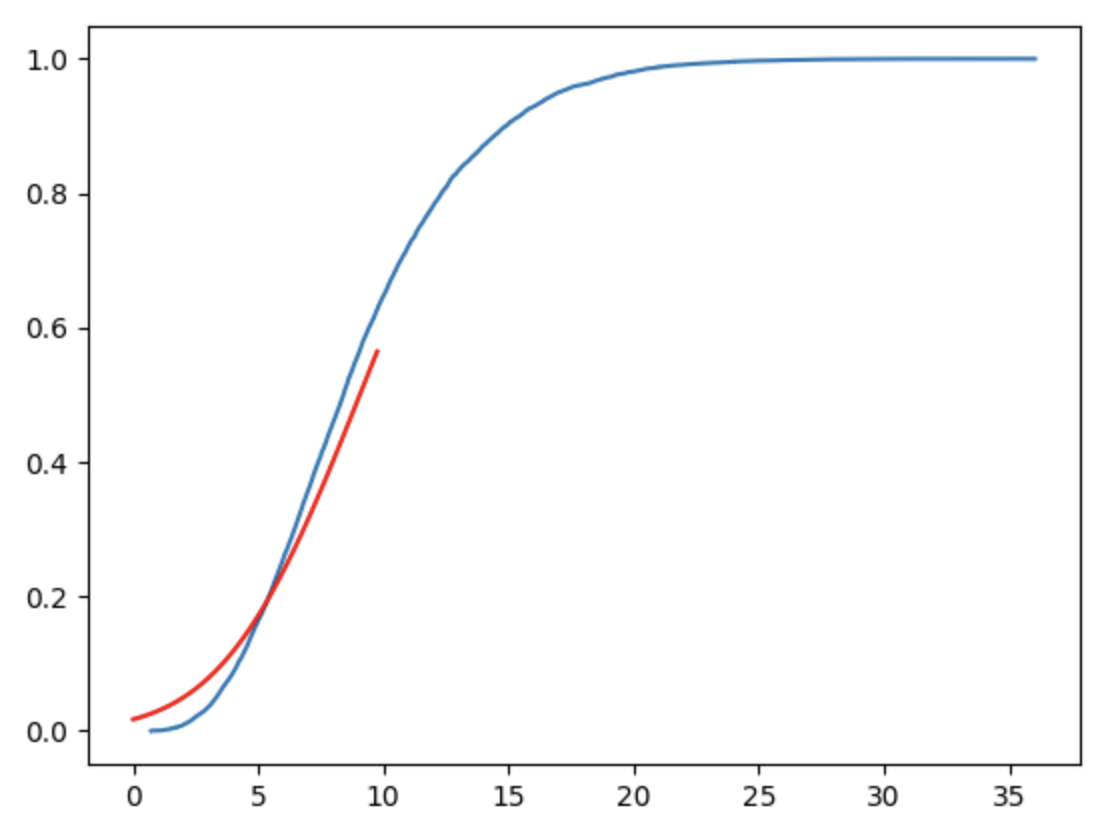
**import** numpy **as** np  
**from** matplotlib **import** pyplot **as** plt  
**from** scipy.stats **import** norm  
**import** math

**def** picture(r, n):  
 *# 1 график* plt.hist(r, density=**True**, label=**"Histogram"**)  
 x = np.arange(0, 30, 0.25)  
 y = []  
 **for** i **in** range(len(x)):  
 y.append(norm.pdf(x[i], np.mean(r), np.std(r)))  
  
 plt.plot(x, y, color=**'r'**)  
 plt.show()  
  
 *# 2 график* tmp = [i/n **for** i **in** range(1, n)]  
 plt.plot(sorted(r), tmp)  
 x = np.arange(0, 10, 0.25)  
 y = norm.cdf(x, np.mean(r), np.std(r))  
 plt.plot(x, y, color=**'r'**)  
 plt.show()  
**def** alg\_2():  
 a1 = np.random.uniform(0, 1, 1)  
 a2 = np.random.uniform(0, 1, 1)  
 **return** math.sqrt(-2 \* math.log(a2)) \* math.cos(2 \* math.pi \* a1)  
  
**def** chi\_2(N):  
 u = []  
 **for** i **in** range(1, N):  
 u.append(alg\_2())  
 **return** sum([i \*\* 2 **for** i **in** u])  
  
n = 10000  
N = 10  
r = []  
**for** i **in** range(1, n):  
 r.append(chi\_2(N))  
  
M = sum(r) / n  
print(**"M: "**, M)  
print(**"D: "**, sum(R \*\* 2 **for** R **in** r) / n - M \*\* 2)  
  
picture(r, n)

Результаты



Графики

# Распределение Стьюдента

Код

**import** numpy **as** np  
**from** matplotlib **import** pyplot **as** plt  
**from** scipy.stats **import** norm  
**import** math

**def** picture(r, n):  
 *# 1 график* plt.hist(r, density=**True**, label=**"Histogram"**)  
 x = np.arange(-5, 5, 0.25)  
 y = []  
 **for** i **in** range(len(x)):  
 y.append(norm.pdf(x[i], np.mean(r), np.std(r)))  
  
 plt.plot(x, y, color=**'r'**)  
 plt.show()  
  
 *# 2 график* tmp = [i/n **for** i **in** range(1, n)]  
 plt.plot(sorted(r), tmp)  
 x = np.arange(0, 10, 0.25)  
 y = norm.cdf(x, np.mean(r), np.std(r))  
 plt.plot(x, y, color=**'r'**)  
 plt.show()

**def** alg\_2():  
 a1 = np.random.uniform(0, 1, 1)  
 a2 = np.random.uniform(0, 1, 1)  
 **return** math.sqrt(-2 \* math.log(a2)) \* math.cos(2 \* math.pi \* a1)  
  
**def** chi\_2(N):  
 u = []  
 **for** i **in** range(1, N):  
 u.append(alg\_2())  
 **return** sum([i \*\* 2 **for** i **in** u])  
  
**def** student(N):  
 **return** alg\_2()/math.sqrt(chi\_2(N)/N)  
  
N = 10  
n = 10000  
r = []  
**for** i **in** range(1, n):  
 r.append(student(N))  
  
M = sum(r) / n  
print(**"M: "**, M)  
print(**"D: "**, sum(R \*\* 2 **for** R **in** r) / n - M \*\* 2)  
  
picture(r, n)

Результаты



Графики

