

## ПРАКТИЧЕСКАЯ ЧАСТЬ 2 МОДУЛЯ КУРСА «МАТЕМАТИЧЕСКАЯ СТАТИСТИКА»

### Повторение

#### *1. Основные термины*

- Случайная величина
- Событие
- Вероятность
- Распределение вероятностей
- Функция вероятности
- Кумулятивная функция распределения
- Выборка
- Математическое ожидание (мат. ожидание, среднее значение)
- Медиана
- Мода
- Дисперсия

#### *2. Предельные теоремы*

- Закон больших чисел (ЗБЧ)
- Центральная предельная теорема (ЦПТ)
- Теорема Муавра-Лапласа (ТМЛ)

#### *3. Методы статистики*

- Статистика
- Классификация методов статистики: параметрические, непараметрические
- Метод моментов
- Метод максимального правдоподобия

### Краткое описание модуля

- Повторение классификаций распределений: дискретные, абсолютно непрерывные
- Изучение основных дискретных распределений
- Изучение основных абсолютно непрерывных распределений
- Знакомство с модулем `scipy` для статистики

## Дискретные распределения

**Решение задачи на дискретное распределение случайной величины  $\xi$  в общем виде:**

- **Массовая функция вероятности:**

<b>Вероятность</b>	<b><math>p_1</math></b>	<b><math>p_2</math></b>	<b>...</b>	<b><math>p_n</math></b>
<b>Значение</b>	<b><math>x_1</math></b>	<b><math>x_2</math></b>	<b>...</b>	<b><math>x_n</math></b>

где  $p_i$  – вероятность наступления события  $\xi=x_i$

- **Условие нормировки:**  $\sum_i p_i = 1$
- **Кумулятивная функция распределения:**

<b>Вероятность</b>	<b><math>P_1</math></b>	<b><math>P_2</math></b>	<b>...</b>	<b>1</b>
<b>Значение</b>	<b><math>x_1</math></b>	<b><math>x_2</math></b>	<b>...</b>	<b><math>x_n</math></b>

где  $P_i$  – вероятность наступления события  $\xi \leq x_i$

- **Среднее значение:**  $\mu = \sum_i p_i \cdot x_i$
- **Дисперсия:**  $\sigma^2 = \sum_i p_i \cdot x_i^2 - \mu^2$ ,  $\sigma$  – среднеквадратичное отклонение

### 1. Равномерное дискретное распределение

**Задача.** В корзине 9 красных шариков и один белый. Наугад вынимают по одному шарiku до тех пор, пока шар не окажется белым. Красные шарики в корзину не возвращаются. Пусть случайная величина  $\xi$  – число красных шариков, покинувших корзину. Найти закон распределения  $\xi$ , математическое ожидание, дисперсию и среднеквадратическое отклонение  $\xi$ , и вероятности следующих событий: белый шарик извлечен за 5 попыток; не менее чем за 3 попытки; число попыток от 7 до 10 включительно.

**Решение.**

$$p_1 = P(\xi=0) = P(\text{сразу вынули белый шар}) = 1/10 = 0.1$$

$$p_2 = P(\xi=1) = P(\text{сначала вынули красный шар, потом белый}) = 9/10 * 1/9 = 0.1$$

$$p_3 = P(\xi=2) = P(\text{сначала вынули два красных шара, потом белый}) = 9/10 * 8/9 * 1/8 = 0.1$$

и т.д.

Таким образом,  $p_{k+1} = P(\xi=k) = p = 0.1 \rightarrow$  случайная величина  $\xi$  принимает целочисленные значения от 0 до 9 с одинаковой вероятностью  $p = 0.1 \rightarrow \xi$  имеет дискретное равномерное распределение с  $n = 10$

**Ряд распределения случайной величины  $\xi$ :**

<b><math>\xi</math></b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>...</b>	<b>k</b>	<b>...</b>	<b>9</b>
<b>p</b>	<b>0.1</b>	<b>0.1</b>	<b>0.1</b>	<b>...</b>	<b>0.1</b>	<b>...</b>	<b>0.1</b>

$$M\xi = (0+9)/2 = 4.5 \text{ (мат. ожидание)}$$

$$D\xi = (100 - 1)/12 = 8.25 \text{ (дисперсия)}$$

$$\sigma\xi = 8.25^{0.5} \approx 2.9 \text{ (среднеквадратическое отклонение)}$$

$$P(\xi=4) = 0.1 \text{ (вероятность, что белый шарик извлечен за 5 попыток)}$$

$$P(\xi \geq 2) = 1 - P(\xi < 2) = 1 - P(\xi=0) - P(\xi=1) = 1 - 0.2 = 0.8 \text{ (вероятность, что белый шарик извлечен не менее чем за 3 попытки)}$$

$$P(6 \leq \xi \leq 9) = P(\xi=6) + P(\xi=7) + P(\xi=8) + P(\xi=9) = 0.4 \text{ (вероятность, что белый шарик извлечен число попыток от 7 до 10 включительно)}$$

### 2. Биномиальное распределение

**Задача 1.** Из  $m=200$  аккумуляторов за год хранения  $l=13$  выходит из строя. Наудачу выбирают  $n=12$  аккумуляторов. Определить вероятность того, что среди них  $k=4$  неисправных. Определить вероятность того, что среди выбранных аккумуляторов неисправно от 1 до 3 включительно.

**Решение.**

Имеем схему Бернулли с параметрами  $p=l/m=0.065$  (вероятность того, что аккумулятор выйдет из строя),  $n=12$  (число испытаний),  $k=4$  (число «успехов», неисправных аккумуляторов).

Будем использовать формулу Бернулли (вероятность того, что в  $n$  испытаниях событие произойдет  $k$  раз):

$$\frac{n!}{k!(n-k)!} p^k q^{n-k}$$

$$P_{12}(4) = 12! / (4! * 8!) * 0.065^4 * 0.935^8 \approx 0.00516$$

Вероятность того, что неисправно от 1 до 3 включительно означает, что неисправно 1, 2 или 3; тогда:

$$P(1 \leq \xi \leq 3) = P_{12}(1) + P_{12}(2) + P_{12}(3) \approx 0.5478$$

Однако эту задачу можно решить, используя Python

```
import numpy as np
from scipy.stats import binom
```

```

import matplotlib.pyplot as plt
plt.figure(figsize=(8,5))
n, p = 13, 0.065
x = np.arange(0, n, 1)
our_pmf = binom.pmf(x, n-1, p)
s = {}
for i in range(n):
    s[i] = round(our_pmf[i], 5)
print(s)
plt.scatter(x, our_pmf)
plt.title('PMF')
plt.grid()
plt.show()
plt.figure(figsize=(8,5))
x = np.arange(0, n, 0.01)
plt.scatter(x, binom.cdf(x, n, p))
plt.title('CDF')
plt.grid()
plt.show()

```

Задача 2. Монеты подбрасывают 10 раз. Случайная величина  $\xi$  – число появления герба в 10 бросаниях монеты. Для случайной величины  $\xi$  найти массовую функцию вероятности (PMF) и кумулятивную функцию распределения (CDF), вычислить математическое ожидание и дисперсию.

```

import numpy as np
from scipy.stats import binom
import matplotlib.pyplot as plt
plt.figure(figsize=(8,5))
n, p = 11, 0.5
x = np.arange(0, n, 1)
our_pmf = binom.pmf(x, n-1, p)
s = {}
for i in range(n):
    s[i] = round(our_pmf[i], 3)
print(s)
plt.scatter(x, our_pmf)
plt.title('PMF')
plt.grid()
plt.show()
plt.figure(figsize=(8,5))
x = np.arange(0, n, 0.01)
plt.scatter(x, binom.cdf(x, n, p))
plt.title('CDF')
plt.grid()
plt.show()
mean, var, skew, kurt = binom.stats(n, p, moments='mvsk')
print(f'Математическое ожидание = {mean}, дисперсия = {var}')

```

### 3. Геометрическое распределение

Задача 1. В корзине 9 красных шариков и один белый. Наугад вынимают по одному шарiku до тех пор, пока шар не окажется белым. Красные шарики возвращаются в корзину. Пусть случайная величина  $\xi$  – это число извлеченных красных шариков, а  $\eta$  – общее число испытаний. Найти законы распределения и вероятностные характеристики случайных величин и вероятности следующих событий: белый шарик извлекли за 5 попыток; не менее чем за 3 попытки.

#### Решение.

Случайная величина  $\xi$  равна числу “неудач” до первого “успеха”, под “неудачей” понимается извлечение красного шара, под “успехом” – белого. Вероятность извлечь белый шар не зависит от номера попытки, поскольку красные шары возвращаются в корзину, то есть  $\xi \sim \text{Geom}(0.1)$ , где  $p = 0.1$  – вероятность появления белого шарика:

$$P(\xi=k) = 0.1 * 0.9^k.$$

Тогда:

$M\xi = 9$  (мат. ожидание)

$D\xi = 90$  (дисперсия)

$\sigma\xi \approx 9.5$  (среднеквадратическое отклонение)

Случайная величина  $\eta$  равна общему числу испытаний, которое совпадает с номером первого “успеха”.

То есть  $\eta \sim \text{Geom}(0.1)$ . В свою очередь вероятность появления белого шарика вычисляется по формуле:

$P(\eta=r) = 0.1 * 0.9^{r-1}$

Тогда:

$M\eta = 10$  (мат. ожидание)

$D\eta = 90$  (дисперсия)

$\sigma\eta \approx 9.5$  (среднеквадратическое отклонение)

Вероятность извлечь белый шарик с 5 попытки:  $P(\eta=5) = 0.1 * 0.9^4 = 0.06561$

Вероятность извлечь белый шарик не менее чем за 3 попытки:  $P(\eta \geq 3) = 1 - P(\eta < 3) = 1 - P(\eta=1) - P(\eta=2) = 1 - 0.1 - 0.1 * 0.9 = 0.81$

Для случайной величины  $\eta$  можно построить модель в Python

```
import numpy as np
from scipy.stats import geom
import matplotlib.pyplot as plt
plt.figure(figsize=(8,5))
n = 20
p = 0.1
x = np.arange(0, n, 1)
our_pmf = geom.pmf(x, p)
s = {}
for i in range(n):
    s[i] = round(our_pmf[i], 5)
print(s)
plt.scatter(x, our_pmf)
plt.title('PMF')
plt.grid()
plt.show()
mean, var, skew, kurt = geom.stats(p, moments='mvsk')
print(f'Математическое ожидание = {mean}, дисперсия = {var}')
```

Задача 2. Студент едет в другой конец города, стоя в переполненном автобусе. Он пытается занять свободное место. Будем считать, что эти попытки не зависят друг от друга, и вероятность успеть к свободному месту постоянна и равна 0.05. Пусть случайная величина  $\xi$  – общее число попыток сесть (последняя – удачная). Найдите закон распределения  $\xi$ , ее вероятностные характеристики и вероятности следующих событий: студенту повезет на 7-й раз; число неудачных попыток не превышает 2; номер успешной попытки начинается с 5.

Решение.

```
import numpy as np
from scipy.stats import geom
import matplotlib.pyplot as plt
plt.figure(figsize=(8,5))
n = 40
p = 0.05
x = np.arange(0, n, 1)
our_pmf = geom.pmf(x, p)
s = {}
for i in range(n):
    s[i] = round(our_pmf[i], 5)
print(s)
plt.scatter(x, our_pmf)
plt.title('PMF')
plt.grid()
plt.show()
mean, var, skew, kurt = geom.stats(p, moments='mvsk')
print(f'Математическое ожидание = {mean}, дисперсия = {round(var, 1)}')
print(f'Вероятность, что повезет на 7 раз = {s[7]}')
print(f'Вероятность, что число неудач не больше 2 = {s[1] + s[2] + s[3]}')
```

```
print(f'Вероятность, что успех будет после 4 попыток = {1 - s[1] - s[2] - s[3] - s[4]}')
```

#### 4. Отрицательное биномиальное распределение (распределение Паскаля)

**Задача 1.** Спутник сканирует заданную акваторию океана за 4 оборота Земли. Если на каком-либо витке из-за различных помех происходит искажение текущего результата, то оно обнаруживается, и сканирование, выполненное на этом витке, повторяется заново. Найти вероятность того, что всё сканирование будет завершено не более чем за 10 витков, если вероятность искажения результата на одном витке составляет 0.2.

##### Решение.

В данном случае случайная величина – вероятность  $k$  неудачных испытаний вплоть до  $r$ -ого успеха (включая этот успех). Поэтому используем распределение Паскаля (а не биномиальное распределение, где случайная величина – это вероятность  $r$  успехов в  $n$  испытаниях).

Неискажённый результат – «успех»,  $p = 0.8$ . Тогда числу  $r$  будет соответствовать число успешно завершённых витков с вероятностью успеха  $p$ , то есть 4 оборота. При этом мы должны добиться этого не более, чем за  $n = 10$  витков. Тогда искомая вероятность будет  $P(\xi \leq 10-4)$ . Используем формулу для  $P(\xi=k)$ :

$$\frac{(k+r-1)!}{k!(r-1)!} p^r q^k$$

Используем Python

```
import numpy as np
from scipy.stats import nbinom
import matplotlib.pyplot as plt
plt.figure(figsize=(8,5))
n, p = 4, 0.8
x = np.arange(0, 12, 1)
our_pmf = nbinom.pmf(x, n, p)
s = {}
for i in range(12):
    s[i] = round(our_pmf[i], 5)
print(s)
plt.scatter(x, our_pmf)
plt.title('PMF')
plt.grid()
plt.show()
print(f'Вероятность, что будет не более 6 неудач (то есть из 10 испытаний 4 хорошие) = {sum(our_pmf[0:7])}')
```

**Задача 2.** Для медицинского исследования последствий от COVID-19 набирается группа из 20 испытуемых. Каждый человек, с которым исследователь проводит собеседование, имеет 60%-ный шанс на участие в исследовании. Какова вероятность того, что исследователю придется опросить более 40 человек?

##### Решение.

«Успех» - участие в исследовании,  $p = 0.6$ . Тогда числу  $r$  будет соответствовать число успешно набранных людей с вероятностью успеха  $p$ , то есть 20. При этом нам придется опросить более 40 человек. Тогда искомая вероятность будет  $1 - P(\xi \leq 40-20)$ .

```
import numpy as np
from scipy.stats import nbinom
import matplotlib.pyplot as plt
plt.figure(figsize=(8,5))
n, p = 20, 0.6
x = np.arange(0, 30, 1)
our_pmf = nbinom.pmf(x, n, p)
s = {}
for i in range(30):
    s[i] = round(our_pmf[i], 5)
print(s)
plt.scatter(x, our_pmf)
plt.title('PMF')
plt.grid()
plt.show()
print(f'Вероятность, что будет более 20 неудач (то есть опросим более 40 человек при нужных 20) = {1 - sum(our_pmf[0:21])}')
```

Задача 1. Организация пытается создать группу из 8 человек, обладающих определенными знаниями о производственном процессе. В организации работают 30 сотрудников, обладающих необходимыми знаниями, причем 10 из них работают в конструкторском бюро. Какова вероятность того, что в группу попадут один или два сотрудника из конструкторского бюро, если членов группы выбирают случайно? Объем генеральной совокупности этой задаче  $N = 30$ , объем выборки  $n = 8$ , а количество успехов  $D = 10$ .

Решение.

Используем формулу для гипергеометрического распределения  $P(\xi=k)$ :

$$\frac{C_D^k C_{N-D}^{n-k}}{C_N^n}$$

В нашем случае  $k = 1$  или  $k=2$ , то есть мы ищем  $P(\xi=1)+P(\xi=2)$ .

Вспользуемся Python.

```
import numpy as np
from scipy.stats import hypergeom
import matplotlib.pyplot as plt
plt.figure(figsize=(8,5))
[N, D, n] = [30, 10, 8]
rv = hypergeom(N, D, n)
x = np.arange(0, D+1)
our_pmf = rv.pmf(x)
s = {}
for i in range(D+1):
    s[i] = round(our_pmf[i], 5)
print(s)
plt.scatter(x, our_pmf)
plt.title('PMF')
plt.grid()
plt.show()
print(f'Вероятность, что попадут один или два сотрудника из КБ = {sum(our_pmf[1:3])}')
```

Задача 2. Для исследования в стае из 50 редких птиц окольцевали 20 особей. Через некоторое время отловили 10 птиц. Для случайной величины  $\xi$  – числа окольцованных птиц среди отловленных, найти массовую функцию вероятности (PMF) и кумулятивную функцию распределения (CDF), с использованием Matplotlib нарисовать их графики, вычислить мат. ожидание и дисперсию.

```
import numpy as np
from scipy.stats import hypergeom
import matplotlib.pyplot as plt
plt.figure(figsize=(8,5))
[N, D, n] = [50, 20, 10]
rv = hypergeom(N, D, n)
x = np.arange(0, D+1)
our_pmf = rv.pmf(x)
s = {}
for i in range(D+1):
    s[i] = round(our_pmf[i], 5)
print(s)
plt.scatter(x, our_pmf)
plt.title('PMF')
plt.grid()
plt.show()
x = np.arange(0, n+1)
plt.scatter(x, rv.cdf(x))
plt.title('CDF')
plt.grid()
plt.show()
mean, var, skew, kurt = hypergeom.stats(N, D, n, moments='mvsk')
print(f'Математическое ожидание = {mean}, дисперсия = {var}')
```

## 6. Распределение Пуассона

Задача 1. Студент делает в среднем 1 ошибку на 5 страниц конспекта. Пусть  $\xi$  – число ошибок, которое студент сделает в 24-листовой тетради. Найти закон распределения  $\xi$ , математическое ожидание, дисперсию,

среднеквадратическое отклонение  $\xi$  и вероятности следующих событий: в конспекте будет 5 ошибок; не меньше 5 ошибок; от 7 до 10 ошибок включительно.

Решение.

Случайная величина  $\xi$  является пуассоновской с параметром  $\lambda$ , т. к. равна числу ошибок, возникающих с постоянной интенсивностью независимо одна от другой.

Вычислим интенсивность появления ошибок в 24-листовой тетради, если в среднем на 5 страницах допускается одна ошибка, а в тетради 48 страниц:  $\lambda = 9.6$ .

Закон распределения  $P(\xi=k)$ :

$$\frac{\lambda^k}{k!} e^{-\lambda}$$

Вероятностные характеристики  $\xi$ :

$M\xi = 9.6$  (мат. ожидание)

$D\xi = 9.6$  (дисперсия)

$\sigma\xi = 9.6^{0.5} \approx 3.1$  (среднеквадратическое отклонение)

$P(\xi=5) \approx 0.046$  (вероятность, что в конспекте 5 ошибок)

$P(\xi \geq 5) = 1 - P(\xi < 5) = 1 - P(\xi=0) - P(\xi=1) - P(\xi=2) - P(\xi=3) - P(\xi=4) \approx 0.9622$  (вероятность, что не меньше 5 ошибок)

$P(7 \leq \xi \leq 10) = P(\xi=7) + P(\xi=8) + P(\xi=9) + P(\xi=10) \approx 0.4755$  (вероятность, что от 7 до 10 ошибок включительно)

Та же задача в Python:

```
import numpy as np
from scipy.stats import poisson
import matplotlib.pyplot as plt
plt.figure(figsize=(8,5))
mu = 9.6
x = np.arange(0, 25, 1)
our_pmf = poisson.pmf(x, mu)
s = {}
for i in range(25):
    s[i] = round(our_pmf[i], 5)
print(s)
plt.scatter(x, our_pmf)
plt.title('PMF')
plt.grid()
plt.show()
print(f'Вероятность, что в конспекте 5 ошибок = {s[5]}')
print(f'Вероятность, что в конспекте не меньше 5 ошибок = {1 - s[0] - s[1] - s[2] - s[3] - s[4]}')
print(f'Вероятность, что в конспекте от 7 до 10 ошибок включительно = {s[7] + s[8] + s[9] + s[10]}')
```

Задача 2. На концерте популярного исполнителя аппаратура ломается с вероятностью 3%, и приходится петь вживую. Пусть  $\xi$  – число концертов, на которых аппаратура вышла из строя. На год запланировано ровно 100 концертов. Найти вероятность того, что аппаратура выйдет из строя не менее чем на четырех концертах.

Решение.

В задаче рассматривается схема Бернулли с малой вероятностью “успеха”  $p = 0.03$ , большим числом испытаний  $n = 100$  и числом “успехов”  $k \ll n$ , где под “успехом” понимается отказ аппаратуры. При заданных условиях вычислить биномиальные коэффициенты и возвести величину  $q = 1-p$  в большую степень достаточно сложно. В таком случае справедлива теорема:

*Если вероятность “успеха”  $p$  в каждом испытании постоянна и мала, а число независимых испытаний  $n$  достаточно велико, то вероятность наступления “успеха” ровно  $k$  раз,  $k \ll n$ , приближенно вычисляется по формуле Пуассона с параметром  $\lambda = np$ . Поэтому распределение Пуассона иногда называют распределением редких событий.*

Тогда в нашем случае  $\lambda = 3$ .

Вероятность отказа аппаратуры не меньше, чем на 4-х концертах:

$P(\xi \geq 4) = 1 - P(\xi < 4) = 1 - P(\xi=0) - P(\xi=1) - P(\xi=2) - P(\xi=3) \approx 0.353$

Сравним ответ при решении через биномиальное распределение и через Пуассона в Python

```
import numpy as np
from scipy.stats import poisson
import matplotlib.pyplot as plt
plt.figure(figsize=(8,5))
```

```

mu = 3
x = np.arange(0, 12, 1)
our_pmf = poisson.pmf(x, mu)
s = {}
for i in range(12):
    s[i] = round(our_pmf[i], 5)
print(s)
plt.scatter(x, our_pmf)
plt.title('PMF')
plt.grid()
plt.show()
y = 1 - s[0] - s[1] - s[2] - s[3]
print(f'Вероятность отказа аппаратуры не меньше, чем на 4-х концертах по Пуассону = {round(y, 4)}')

```

```

import numpy as np
from scipy.stats import binom
import matplotlib.pyplot as plt
plt.figure(figsize=(8,5))
n, p = 100, 0.03
x = np.arange(0, n, 1)
our_pmf = binom.pmf(x, n-1, p)
s = {}
for i in range(12):
    s[i] = round(our_pmf[i], 5)
print(s)
plt.scatter(x[:12], our_pmf[:12])
plt.title('PMF')
plt.grid()
plt.show()
z = 1 - s[0] - s[1] - s[2] - s[3]
print(f'Вероятность отказа аппаратуры не меньше, чем на 4-х концертах по Биномиальному = {round(z, 4)}')

print(abs(1-y/z))

```

В итоге получили относительную погрешность 2%. То есть теорема действительно неплохо работает.

## 7. Произвольное дискретное распределение

**Задача.** На переэкзаменовку по теории вероятностей явились 3 студента. Вероятность того, что студент сдаст экзамен, равна 0,8. Найдите ряд распределения случайной величины  $\xi$  числа студентов, сдавших экзамен, найдите  $M(\xi)$ ,  $D(\xi)$ .

**Решение.**

Введем независимые события:

$A_1$  = (Первый студент сдал экзамен)

$A_2$  = (Второй студент сдал экзамен)

$A_3$  = (Третий студент сдал экзамен)

Вероятности известны:  $P(A_1) = P(A_2) = P(A_3) = 0.8$ . Соответственно, вероятности того, что событие не произойдет, равны:  $P(\neg A_1) = P(\neg A_2) = P(\neg A_3) = 0.2$ .

$p_0 = P(\xi=0) = P(\text{ни один не сдал экзамен}) = P(\neg A_1) * P(\neg A_2) * P(\neg A_3) = 0.008$

$p_1 = P(\xi=1) = P(\text{один из трех сдал}) = P(A_1) * P(\neg A_2) * P(\neg A_3) + P(\neg A_1) * P(A_2) * P(\neg A_3) + P(\neg A_1) * P(\neg A_2) * P(A_3) = 0.096$

$p_2 = P(\xi=2) = P(\text{двое из трех сдали}) = P(A_1) * P(A_2) * P(\neg A_3) + P(\neg A_1) * P(A_2) * P(A_3) + P(A_1) * P(\neg A_2) * P(A_3) = 0.384$

$p_3 = P(\xi=3) = P(\text{все трое сдали}) = P(A_1) * P(A_2) * P(A_3) = 0.512$

Ряд распределения случайной величины  $\xi$ :

$\xi$	0	1	2	3
p	0.008	0.096	0.384	0.512

$$M\xi = 0*p_0 + 1*p_1 + 2*p_2 + 3*p_3 = 2.4$$

$$D\xi = 0^2*p_0 + 1^2*p_1 + 2^2*p_2 + 3^2*p_3 - 2.4^2 = 0.48$$



```

import numpy as np
import matplotlib.pyplot as plt
plt.figure(figsize=(8,5))
s = {0: 0.008,
     1: 0.096,
     2: 0.384,
     3: 0.512}
plt.scatter(s.keys(), s.values())
plt.title('PMF')
plt.grid()
plt.show()
our_cdf = [0, 0, 0, 0, 0]
our_pmf = [0.008, 0.096, 0.384, 0.512]
for i in range(1, len(our_pmf)+1):
    our_cdf[i] = our_pmf[i-1] + our_cdf[i-1]
x = np.arange(-0.5, 3.5, 0.001)
F = np.zeros(len(x))
for i in range(len(x)):
    if x[i]<=0: F[i] = our_cdf[0]
    if 0<x[i]<=1: F[i] = our_cdf[1]
    if 1<x[i]<=2: F[i] = our_cdf[2]
    if 2<x[i]<=3: F[i] = our_cdf[3]
    if 3<x[i]: F[i] = our_cdf[4]
plt.scatter(x, F)
plt.title('CDF')
plt.grid()
plt.show()
M = 0
D = 0
for i in range(len(our_pmf)):
    M = M+our_pmf[i]*i
    D = D+our_pmf[i]*i*i
D = D - M*M
print(f'Математическое ожидание = {M}, дисперсия = {round(D, 2)}')

```

## 8. Работа с дискретными распределениями в pandas

```

import numpy as np
import pandas as pd
from scipy import stats
import matplotlib.pyplot as plt
S = 10000
x = range(0, S, 1)
r1 = stats.randint.rvs(low = 15, high = 30, size=S)
r2 = stats.binom.rvs(n=20, p=0.3, size=S)
r3 = stats.geom.rvs(p = 0.4, size=S)
r4 = stats.nbinom.rvs(n=25, p=0.6, size=S)
r5 = stats.hypergeom.rvs(M=300, n=100, N=150, size=S)
r6 = stats.poisson.rvs(mu=0.8, size=S)

df = pd.DataFrame()
df = pd.DataFrame()
df['randint'] = r1
df['binom'] = r2
df['geom'] = r3
df['nbinom'] = r4
df['hypergeom'] = r5
df['poisson'] = r6

```

```
df.hist(bins=15)
```

```
for i in df.columns:  
    df[i].hist(bins=15)
```

```
sl = {}  
for i in df.columns:  
    sl[i] = (df[i].mean(), df[i].var())  
sl
```

## Непрерывные распределения

Решение задачи на непрерывное распределение случайной величины  $\xi$  в общем виде:

- Функция плотности вероятности:  $f(x)$  (сравнительная вероятность реализации событий)
- Условие нормировки:  $\int_{-\infty}^{+\infty} f(x)dx = 1$
- Кумулятивная функция распределения:  $F(x_0) = \int_{-\infty}^{x_0} f(x)dx$  (вероятность события  $\xi \in (-\infty; x_0]$ )
- Среднее значение:  $\mu = \int_{-\infty}^{+\infty} x \cdot f(x)dx$
- Дисперсия:  $\sigma^2 = \int_{-\infty}^{+\infty} x^2 \cdot f(x)dx - \mu^2$ ,  $\sigma$  – среднеквадратичное отклонение

### 1. Равномерное непрерывное распределение

Задача 1. Цена деления шкалы измерительного прибора равна 0,2. Показания прибора округляют до ближайшего целого деления. Найти вероятность того, что при отсчете будет сделана ошибка: А) меньшая 0,04; Б) большая 0,05.

Решение.

Введем случайную величину  $X$  - ошибку округления, которая распределена равномерно в интервале между двумя соседними целыми делениями. Так как в задаче цена деления равна 0,2, плотность распределения:

$$f(x) = \begin{cases} 0, & x \leq 0, \\ \frac{1}{0,2}, & 0 < x \leq 0,2, \\ 0, & x > 0,2. \end{cases} = \begin{cases} 0, & x \leq 0, \\ 5, & 0 < x \leq 0,2, \\ 0, & x > 0,2. \end{cases}$$

А) Найдем вероятность того, что при отсчете будет сделана ошибка меньшая 0,04, то есть  $X$  попадет в интервал  $(0; 0,04)$  или  $(0,16; 0,2)$ , вероятность этого:

$$\begin{aligned} P(0 < X < 0,04) + P(0,16 < X < 0,2) &= \int_0^{0,04} 5dx + \int_{0,16}^{0,2} 5dx = 5x \Big|_0^{0,04} + 5x \Big|_{0,16}^{0,2} = \\ &= 5 \cdot 0,04 + 5 \cdot (0,2 - 0,16) = 5 \cdot 0,08 = 0,4. \end{aligned}$$

Б) Найдем вероятность того, что при отсчете будет сделана ошибка большая 0,05, то есть  $X$  попадет в интервал  $(0,05; 0,15)$ , вероятность этого:

$$P(0,05 < X < 0,15) = \int_{0,05}^{0,15} 5dx = 5x \Big|_{0,05}^{0,15} = 5 \cdot (0,15 - 0,05) = 5 \cdot 0,1 = 0,5.$$

Задача 2. Автобусы идут с интервалом 5 минут. Полагая, что случайная величина  $\xi$  – время ожидания автобуса на остановке – распределена равномерно на указанном интервале, найти среднее время ожидания и среднеквадратическое отклонение времени ожидания.

Решение.

Пусть случайная величина  $\xi$  распределена равномерно на интервале  $[a, b] = [0, 5]$ . Плотность распределения этой случайной величины:

$$f(x) = \begin{cases} 0, & x < 0, \\ A, & 0 \leq x \leq 5, \\ 0, & x > 5. \end{cases}$$

Найдем параметр  $A$  из условия нормировки:

$$\int_{-\infty}^{+\infty} f(x)dx = \int_0^5 A dx = Ax \Big|_0^5 = 5A = 1$$

Отсюда  $A = 1/5$ .

Среднее время ожидания на остановке – это математическое ожидание случайной величины  $\xi$ , то есть:

$$M(\xi) = \int_{-\infty}^{+\infty} f(x)x dx = \int_0^5 \frac{x}{5} dx = \frac{1}{10} x^2 \Big|_0^5 = \frac{25}{10} = \frac{5}{2}$$

Найдем дисперсию:

$$D(\xi) = \int_{-\infty}^{+\infty} f(x)x^2 dx - (M(\xi))^2 = \int_0^5 \frac{x^2}{5} dx - \frac{25}{4} = \frac{1}{15} x^3 \Big|_0^5 - \frac{25}{4} = \frac{125}{15} - \frac{25}{4} = \frac{25}{12}$$

Тогда среднее квадратичное отклонение времени ожидания равно корню из дисперсии, то есть  $\approx 1$  минута 26 секунд.

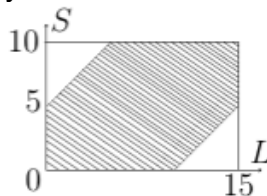
Посмотрим на оформление решения в Python

```
import numpy as np
from scipy.stats import uniform
import matplotlib.pyplot as plt
plt.figure(figsize=(8,5))
x = np.arange(-2, 8, 0.001)
our_pdf = uniform.pdf(x, 0, 5)
our_cdf = uniform.cdf(x, 0, 5)
plt.plot(x, our_pdf)
plt.title('PDF')
plt.grid()
plt.show()
plt.plot(x, our_cdf)
plt.title('CDF')
plt.grid()
plt.show()
mean, var, skew, kurt = uniform.stats(0, 5, moments='mvsk')
print(f'Математическое ожидание = {mean}, дисперсия = {round(var, 4)}')
```

**Задача 3.** Студент Иван заметил, что лектор приходит на занятия со случайным опозданием в пределах 15 мин, при этом он разрешает проходить в аудиторию только тем студентам, которые пришли после него не позднее 5 мин. Иван тоже решил опоздать на лекцию, но выбрал себе границу случайного опоздания всего в 10 мин. и время ожидания лектора тоже 10 мин. Какова вероятность того, что он все же посетит лекцию?

Решение.

Обозначим через  $L$  и  $S$  время опоздания на лекцию лектором и студентом соответственно. Предположим, что пара случайных чисел  $(L; S)$  равномерно распределена в прямоугольнике  $[0; 15] \times [0; 10]$ . Тогда студент попадет на лекцию, если произойдут два события  $L + 5 \geq S$  и  $L \leq S + 10$ . Геометрически это соответствует заштрихованной области на рисунке.



То есть искомая вероятность – это отношение площади заштрихованной области к полной площади возможных значений. Хотя в данном случае площадь легко находится аналитически (ответ будет  $5/6$ ), применим универсальный метод Монте-Карло (метод статистических испытаний).

Поместим фигуру в прямоугольник и будем наугад бросать точки в этот прямоугольник. Будем исходить из того, что чем больше площадь фигуры, тем чаще в нее будут попадать точки. Таким образом, при большом числе  $n$  точек, наугад выбранных внутри прямоугольника, доля точек, содержащихся в данной фигуре  $k$ , приближенно равна отношению площади этой фигуры и площади прямоугольника.

Если площадь прямоугольника равна  $S_0$  и в результате испытаний, из которых при  $k$  исходах случайные точки оказались внутри фигуры, то площадь  $S$  фигуры будет определяться выражением  $S = (n/k) \cdot S_0$ , а искомая вероятность  $P = n/k$ .

```
import random
n = 100000
k = 0
for _ in range(n):
    x = random.uniform(0, 15)
    y = random.uniform(0, 10)
    if x+5 >= y and x <= y+10:
        k += 1
print((k/n))
```

В результате многократных расчетов видно, что численный ответ близок к аналитическому. Можно слегка оптимизировать данный код, получив несколько численных значений и взяв их среднее (фактически воспользоваться центральной предельной теоремой):

```
import random
n = 100000
k = 0
```

```

s = []
m = 50
for i in range(m):
    for _ in range(n):
        x = random.uniform(0, 15)
        y = random.uniform(0, 10)
        if x+5 >= y and x <= y+10:
            k += 1
    s.append(k/m)
    k=0
print((sum(s)/n))

```

## 2. Нормальное распределение

Задача 1. Дневная добыча угля в некоторой шахте распределена по нормальному закону с математическим ожиданием 870 тонн и стандартным отклонением 90 тонн.

А) Найдите вероятность того, что в определенный день будут добыты по крайней мере 900 тонн угля.

Б) Определите долю рабочих дней, в которые будет добыто от 860 до 940 тонн угля.

В) Найдите вероятность того, что в данный день добыча угля окажется ниже 750 тонн.

### Решение.

Используем формулу для нахождения вероятности попадания нормальной случайной величины (дневной добычи угля  $X$ ) в интервал  $(\alpha; \beta)$ :

$$P(\alpha < X < \beta) = \Phi\left(\frac{\beta - a}{\sigma}\right) - \Phi\left(\frac{\alpha - a}{\sigma}\right)$$

Здесь  $\Phi(x)$  – функция Лапласа (значения берутся из таблицы),  $a = 870$  – математическое ожидание,  $\sigma = 90$  – среднее квадратическое отклонение.

А) Найдём вероятность того, что по крайней мере 900 тонн будут добыты в определенный день.

$$P(900 < X < \infty) = \Phi\left(\frac{\infty - 870}{90}\right) - \Phi\left(\frac{900 - 870}{90}\right) = \Phi(\infty) - \Phi(0,33) = 0,5 - 0,1293 = 0,3707.$$

Б) Определим долю рабочих дней, в которые будет добыто от 860 до 940 тонн угля.

$$P(860 < X < 940) = \Phi\left(\frac{940 - 870}{90}\right) - \Phi\left(\frac{860 - 870}{90}\right) = \Phi(0,78) - \Phi(-0,11) = \Phi(0,78) + \Phi(0,11) = 0,2823 + 0,0438 = 0,3261.$$

В) Найдём вероятность того, что в данный день добыча угля окажется ниже 750 тонн.

$$P(0 < X < 750) = \Phi\left(\frac{750 - 870}{90}\right) - \Phi\left(\frac{0 - 870}{90}\right) = \Phi(-1,33) - \Phi(-9,67) = -\Phi(1,33) + \Phi(9,67) = -0,4082 + 0,5 = 0,0918.$$

Задача 2. Станок изготавливает шарики для подшипников. Шарик считается годным, если отклонение  $X$  диаметра шарика от проектного размера по абсолютной величине меньше 350 мкм. При этом сам станок уже износился настолько, что может сделать шарик с максимальным отклонением от проектного размера 600 мкм. Для случайной величины  $X$  найти функцию плотности вероятности (PDF) и кумулятивную функцию распределения (CDF), вычислить мат. ожидание и дисперсию. Найти, сколько в среднем будет годных шариков среди 50 изготовленных.

### Решение.

```

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

plt.figure(figsize=(8,5))
x = np.arange(-700, 700, 1)
plt.plot(x, norm.pdf(x, 0, 200), c="blue", label = 'm=0, s=200')
plt.grid()
plt.show()

plt.figure(figsize=(8,5))
x = np.arange(-700, 700, 1)
plt.plot(x, norm.cdf(x, 0, 200), c="blue", label = 'm=0, s=200')
plt.grid()
plt.show()

P = 0.5*(1+math.erf((350-0)/(200*(2**0.5)))) - 0.5*(1+math.erf((-350-0)/(200*(2**0.5))))
print(f'Вероятность того, что отклонение не превышает 350 мкм, равна {round(P, 2)}')

```

```
print(f'Из 50 изготовленных шариков в среднем годными будут {round(P, 2)*50}')
```

### 3. Экспоненциальное распределение

Задача 1. Длительность телефонного разговора – это случайная величина  $\xi$ , которая подчиняется показательному закону. Известно, что вероятность того, что разговор продлится более 7 минут, равна 0,3. Для случайной величины  $\xi$  найти функцию плотности вероятности и кумулятивную функцию распределения, вычислить математическое ожидание и дисперсию.

#### Решение.

Вероятность попадания в интервал  $(a, b)$  непрерывной случайной величины  $\xi$ , распределенной по показательному закону, находится по формуле:  $P(a < \xi < b) = F(b) - F(a) = e^{-\lambda a} - e^{-\lambda b}$ . В нашем случае  $a = 7$ ,  $b = +\infty$ . Тогда  $0.3 = e^{-7\lambda}$ , отсюда  $\lambda \approx 0.172$ .

Функция плотности вероятности:  $f(x) = 0.172 * e^{-0.172 * x}$ . Функция распределения:  $F(x) = 1 - e^{-0.172 * x}$ . Математическое ожидание  $\approx 5.814$ , дисперсия  $\approx 33.804$ .

Такое же решение через Python

```
import math
a = -math.log(0.3)/7
print('lambda =', round(a, 3))
import numpy as np
from scipy.stats import expon
import matplotlib.pyplot as plt
plt.figure(figsize=(8, 5))
x = np.arange(0, 30, 0.001)
plt.plot(x, expon.pdf(x, 0, 1/a), c="blue", label = 'lambda = a')
plt.grid()
plt.show()
plt.figure(figsize=(8, 5))
x = np.arange(0, 30, 0.001)
plt.plot(x, expon.cdf(x, 0, 1/a), c="blue", label = 'lambda = a')
plt.grid()
plt.show()
mean, var, skew, kurt = expon.stats(moments='mvsk', scale = 1/a)
print(f'Математическое ожидание = {round(mean, 3)}, дисперсия = {round(var, 3)}')
```

Задача 2. Для имеющейся совокупности опытных данных (выборки) требуется:

- Построить статистический ряд и гистограмму распределения.
- Вычислить следующие статистики распределения: выборочную среднюю, выборочное среднеквадратическое отклонение, коэффициент вариации, асимметрию, эксцесс. Раскрыть смысловую сторону каждой статистики.
- Обосновать выбор теоретического распределения и методом моментов найти его параметры.
- Построить теоретическую кривую распределения.

18,4	0,8	4,9	10,5	8,8	24,9	16,0	6,0	4,6	16,2
7,6	16,1	13,1	3,0	23,6	4,7	1,8	5,6	6,2	11,4
0,3	9,2	2,6	11,5	2,6	18,9	3,8	4,9	22,4	7,9
12,2	2,0	2,5	5,2	11,2	15,4	13,6	2,4	10,5	3,4
0,7	3,6	2,2	10,3	63,7	13,9	7,5	35,0	28,6	0,2
7,7	4,0	3,6	10,4	15,6	48,3	7,5	8,0	0,9	20,3
18,2	2,8	0,8	17,4	4,0	0,9	1,9	22,6	3,4	18,1
15,8	3,6	2,7	21,5	30,2	3,9	19,2	9,3	21,8	25,1
0,0	33,6	2,1	7,4	4,6	37,0	10,4	15,0	2,6	5,0
6,6	16,0	6,1	9,9	7,6	1,8	11,2	3,8	21,3	0,5

#### Решение.

```
# открытие документа с данными и их запись в список
with open('table.txt', 'r') as f:
    temp = []
```

```
s = f.readlines()
for i in s:
    for j in i.split():
        temp.append(float(j))
print(temp)
```

```
# расчет максимума и минимума выборки
```

```
import pandas as pd
df = pd.DataFrame(temp)
df.columns = ['sample']
df.hist(bins=13)
print(f'Максимальное значение = {df['sample'].max()}, минимальное значение = {df['sample'].min()}')
```

```
# создание словаря, где ключ - кортеж с диапазоном и средним по диапазону, значение - частота попадания в диапазон
```

```
s = {
    ('0-5', 2.5): df[(0 < df['sample']) & (df['sample'] <= 5)].count()['sample'],
    ('5-10', 7.5): df[(5 < df['sample']) & (df['sample'] <= 10)].count()['sample'],
    ('10-15', 12.5): df[(10 < df['sample']) & (df['sample'] <= 15)].count()['sample'],
    ('15-20', 17.5): df[(15 < df['sample']) & (df['sample'] <= 20)].count()['sample'],
    ('20-25', 22.5): df[(20 < df['sample']) & (df['sample'] <= 25)].count()['sample'],
    ('25-30', 27.5): df[(25 < df['sample']) & (df['sample'] <= 30)].count()['sample'],
    ('30-35', 32.5): df[(30 < df['sample']) & (df['sample'] <= 35)].count()['sample'],
    ('35-40', 37.5): df[(35 < df['sample']) & (df['sample'] <= 40)].count()['sample'],
    ('40-45', 42.5): df[(40 < df['sample']) & (df['sample'] <= 45)].count()['sample'],
    ('45-50', 47.5): df[(45 < df['sample']) & (df['sample'] <= 50)].count()['sample'],
    ('50-55', 52.5): df[(50 < df['sample']) & (df['sample'] <= 55)].count()['sample'],
    ('55-60', 57.5): df[(55 < df['sample']) & (df['sample'] <= 60)].count()['sample'],
    ('60-65', 62.5): df[(60 < df['sample']) & (df['sample'] <= 65)].count()['sample'],
    ('65-70', 67.5): df[(65 < df['sample']) & (df['sample'] <= 70)].count()['sample'],
    ('70-75', 72.5): df[(70 < df['sample']) & (df['sample'] <= 75)].count()['sample'],
    ('75-80', 77.5): df[(75 < df['sample']) & (df['sample'] <= 80)].count()['sample'],
    ('80-85', 82.5): df[(80 < df['sample']) & (df['sample'] <= 85)].count()['sample'],
    ('85-90', 87.5): df[(85 < df['sample']) & (df['sample'] <= 90)].count()['sample'],
    ('90-95', 92.5): df[(90 < df['sample']) & (df['sample'] <= 95)].count()['sample'],
    ('95-100', 97.5): df[(95 < df['sample']) & (df['sample'] <= 100)].count()['sample'],
}
for i in s.keys():
    s[i] = s[i] / len(df)
print(s)
```

```
# расчет моментов
```

```
m, d, v, a, e = 0, 0, 0, 0, 0
```

```
for i in s.keys():
```

```
    m = m + i[1]*s[i]
```

```
    d = d + i[1]*i[1]*s[i]
```

```
    a = a + i[1]*i[1]*i[1]*s[i]
```

```
    e = e + i[1]*i[1]*i[1]*i[1]*s[i]
```

```
e = e - 4*a*m + 6*d*m*m - 3*m*m*m*m
```

```
a = a - 3*m*d + 2*m*m*m
```

```
d = (d - m*m)**0.5
```

```
a = a/d**3
```

```
e = e/d**4 - 3
```

```
v = d/m
```

```
print(f' Выборочное среднее {round(m, 3)}, выборочное отклонение {round(d, 3)}, коэффициент вариации {round(v, 3)}, асимметрия {round(a, 3)}, эксцесс {round(e, 3)}')
```

Выборочная средняя - статистический аналог математического ожидания.

Выборочная дисперсия – аналог дисперсии; характеризует рассеяние случайной величины вокруг её среднего значения.

Коэффициент вариации – показатель разброса значений признака относительно некоторой средней величины.

Коэффициент асимметрии служат для характеристики асимметрии (скошенности) распределения. В нашем случае асимметрия близка к 1.5, что характерно для показательного распределения (для нормального или равномерного распределения, когда распределение симметрично относительно математического ожидания, асимметрия близка к нулю).

Эксцесс характеризует крутость (островершинность или плосковершинность) распределения. Кривые, более островершинные, чем нормальная, обладают положительным эксцессом, более плосковершинные - отрицательным эксцессом. В нашем случае эксцесс распределения положителен, что говорит об островершинности кривой распределения, что в совокупности с большой асимметрией указывает на показательное распределение (для нормального распределения и асимметрия, и эксцесс близки к нулю).

```
# визуализация и сравнение
from scipy.stats import expon
import matplotlib.pyplot as plt
plt.figure(figsize=(8, 5))
x = []
for i in s.keys():
    x.append(i[1])
plt.scatter(x, s.values(), c = 'blue')
x = np.arange(0, 65, 0.001)
plt.plot(x, 5*expon.pdf(x, 0, m), c = 'red')
plt.plot(x, 5*expon.pdf(x, 0, df.mean()), c = 'green')
plt.grid()
plt.show()
```

#### 4. Распределение Парето

Задача. Используя распределение Парето, определить, какое количество налогоплательщиков имеет доход, превышающий необлагаемый минимум  $x_0$  в  $k$  раз. Найти математическое ожидание этого распределения и дисперсию. Известно, что  $\alpha=1.5$ ,  $k=6$ ,  $x_0=20$ .

Решение.

```
import numpy as np
from scipy.stats import pareto
import matplotlib.pyplot as plt
plt.figure(figsize=(8,5))
x = np.arange(15, 40, 0.001)
a, k, x0 = 1.5, 6, 20
plt.plot(x, pareto.pdf(x, a, 0, x0), c="green")
plt.grid()
plt.show()
plt.plot(x, pareto.cdf(x, a, 0, x0), c="green")
plt.grid()
plt.show()
print(f'Доля плательщиков с доходом, выше необлагаемого в 6 раз, равна {round((1/k)**a, 4)}')
print(f'Медиана распределения {pareto.mean(a, 0, x0)}')
```

#### 5. Распределение Рэлея

Задача. Спортсмен стреляет из лука по мишени диаметром 70 см. Опыт спортсмена такой, что максимальные отклонения разброса его выстрелов по абсциссе и ординате от центра одинаковы и равны 24 см. Расстояние от точки попадания стрелы до центра мишени (отклонение разброса) – это случайная величина  $\xi$ . Для случайной величины  $\xi$  найти функцию плотности вероятности (PDF) и кумулятивную функцию распределения (CDF), построить их графики, вычислить мат. ожидание и дисперсию. Считая диаметр «яблочка» мишени 10 см, вычислите вероятность того, что стрелок попадет в него.

Решение.

Расстояния от центра до точек попадания стрел — это случайные величины, имеющие  $X$  и  $Y$  ортогональные составляющие (случайный вектор). Эти случайные величины распределены по нормальному закону, тогда средние квадратические отклонения разброса по абсциссе и ординате одинаковы и равны  $24/3 = 8$  см, а расстояние от точки попадания стрелы до центра мишени (отклонение разброса) будет случайной величиной с распределением Рэлея.

```
import numpy as np
from scipy.stats import rayleigh
import matplotlib.pyplot as plt
plt.figure(figsize=(8,5))
x = np.arange(0, 40, 0.001)
plt.plot(x, rayleigh.pdf(x, 0, 8), c="blue")
plt.grid()
plt.show()
plt.figure(figsize=(8,5))
x = np.arange(0, 40, 0.001)
plt.plot(x, rayleigh.cdf(x, 0, 8), c="blue")
plt.grid()
```



```
plt.show()
mean, var, skew, kurt = rayleigh.stats(moments='mvsk', scale = 8)
print(f'Математическое ожидание = {round(mean, 4)}, дисперсия = {round(var, 4)}')
import math
print('Вероятность попадания в яблочко:', round(1-math.exp(-25/128), 4))
```

Нужно отметить, что если независимые гауссовские случайные величины  $X$  и  $Y$  имеют ненулевые математические ожидания, в общем случае неравные, то распределение Рэлея переходит в распределение Райса.

## 6. Произвольное непрерывное распределение

Задача 1. Известно, что на отрезке  $[0;1]$  задана плотность распределения  $f(x) = 0.2 \cdot (2x+4)$  (на остальном множестве ноль). Найдите среднее значение и дисперсию распределения, постройте графики  $f(x)$  и  $F(x)$ , а также определите вероятность того, что случайное число из распределения лежит в диапазоне  $(0.64; 1)$ .

Решение.

```
import numpy as np
import matplotlib.pyplot as plt
plt.figure(figsize=(8,5))
x = np.arange(-0.5, 1.5, 0.0001)
our_pdf = np.zeros(len(x))
for i in range(len(x)):
    if x[i]<0 or x[i]>1:
        our_pdf[i]=0
    else:
        our_pdf[i]=0.2*(2*x[i]+4)
plt.plot(x, our_pdf)
plt.title('PDF')
plt.grid()
plt.show()
```

```
from scipy.integrate import quad
def integrand(x):
    return 0.2*(2*x + 4)
```

```
our_cdf = np.zeros(len(x))
for i in range (len(x)):
    if x[i]<0:
        our_cdf[i] = 0
    elif x[i]>1:
        our_cdf[i] = 1
    else:
        our_cdf[i] = quad(integrand, 0, x[i])[0]
plt.plot(x, our_cdf)
plt.title('CDF')
plt.grid()
plt.show()
print(f'Вероятность, что случайное число лежит от 0.64 до 1 равна {1-quad(integrand, 0, 0.64)[0]}')
```

```
from scipy.integrate import quad
def integrand_mean(x):
    return 0.2*(2*x + 4)*x
```

```
from scipy.integrate import quad
def integrand_var(x):
    return 0.2*(2*x + 4)*x**2
```

```
mean = quad(integrand_mean, 0, 1)[0]
var = quad(integrand_var, 0, 1)[0] - mean**2
print(f'Математическое ожидание = {round(mean, 4)}, дисперсия = {round(var, 4)}')
```

Задача 2. Известно, что на отрезке  $[0;2]$  задана плотность распределения  $f(x) = A*(\sin(x)+\cos(x))$  (на остальном множестве ноль). Найдите  $A$ , среднее значение и дисперсию распределения, постройте графики  $f(x)$  и  $F(x)$ , а также определите вероятность того, что случайное число из распределения лежит в диапазоне  $(1; 2)$ .

Решение.

```
import numpy as np
import matplotlib.pyplot as plt
import math
```

```
from scipy.integrate import quad
def integrand(x):
    return (math.sin(x) + math.cos(x))
```

```
a = quad(integrand, 0, 2)[0]
print(1/a) # 0.43
```

```
from scipy.integrate import quad
def integrand(x):
    return 0.43*(math.sin(x) + math.cos(x))
```

```
plt.figure(figsize=(8,5))
x = np.arange(-0.5, 2.5, 0.0001)
our_pdf = np.zeros(len(x))
for i in range(len(x)):
    if x[i]<0 or x[i]>2:
        our_pdf[i]=0
    else:
        our_pdf[i]=0.43*(math.sin(x[i]) + math.cos(x[i]))
plt.plot(x, our_pdf)
plt.title('PDF')
plt.grid()
plt.show()
```

```
our_cdf = np.zeros(len(x))
for i in range (len(x)):
    if x[i]<0:
        our_cdf[i] = 0
    elif x[i]>2:
        our_cdf[i] = 1
    else:
        our_cdf[i] = quad(integrand, 0, x[i])[0]
plt.plot(x, our_cdf)
plt.title('CDF')
plt.grid()
plt.show()
print(f'Вероятность, что случайное число лежит от 1 до 2 равна {1-quad(integrand, 1, 2)[0]}')
```

```
from scipy.integrate import quad
def integrand_mean(x):
    return 0.43*(math.sin(x) + math.cos(x))*x
```

```
from scipy.integrate import quad
def integrand_var(x):
    return 0.43*(math.sin(x) + math.cos(x))*x**2
```

```
mean = quad(integrand_mean, 0, 2)[0]
var = quad(integrand_var, 0, 2)[0] - mean**2
```

```
print(f'Математическое ожидание = {round(mean, 4)}, дисперсия = {round(var, 4)}')
```

Задача 3. Плотность распределения системы двух случайных величин (X,Y) задана выражением:

$$f(x, y) = \frac{a}{1 + x^2 + x^2 y^2 + y^2}$$

Определить вероятность попадания случайной точки в прямоугольник с вершинами O(0;0), A(0;1), B(√3;1) и C(√3;0).

Решение.

```
from scipy.integrate import dblquad
f = lambda x, y: 1/(1+x*x+x*x*y*y+y*y)
a = dblquad(f, -np.inf, np.inf, -np.inf, np.inf)[0]
print(1/a) # (1/pi**2)
```

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits import mplot3d
import math
plt.figure(figsize=(8,5))
x = np.arange(-5, 5, 0.01)
y = np.arange(-5, 5, 0.01)
X, Y = np.meshgrid(x, y)
Z = (1/math.pi**2)*1/(1+X*X+X*X*Y*Y+Y*Y)
fig = plt.figure()
ax = plt.axes(projection='3d')
ax.plot_surface(X, Y, Z, cmap='cividis')
ax.set_title('PDF')
```

```
f = lambda x, y: (1/math.pi**2)/(1+x*x+x*x*y*y+y*y)
p = dblquad(f, 0, 3**0.5, 0, 1)[0]
print(f'Вероятность попасть у заданный прямоугольник равна {round(p, 4)}') # 1/12
```

### 7. Работа с непрерывными распределениями в pandas

```
import numpy as np
import pandas as pd
from scipy import stats
import matplotlib.pyplot as plt
S = 10000
x = range(0, S, 1)
r1 = stats.uniform.rvs(loc=100, scale=60, size=S)
r2 = stats.norm.rvs(loc=2, scale=4, size=S)
r3 = stats.expon.rvs(loc=10, scale=4, size=S)
r4 = stats.pareto.rvs(5, 0, 35, size=S)
r5 = stats.rayleigh.rvs(20, 8, size=S)
```

```
df = pd.DataFrame()
df['uniform'] = r1
df['norm'] = r2
df['expon'] = r3
df['pareto'] = r4
df['rayleigh'] = r5
```

```
df.hist(bins=25)
```

```
for i in df.columns:
    df[i].hist(bins=25)
```

```
sl = {}
for i in df.columns:
    sl[i] = (df[i].mean(), df[i].var())
sl
```